BULETINUL ŞTIINŢIFIC

al

Universităţii „POLITEHNICA" din Timişoara, România

Seria ELECTRONICĂ ŞI TELECOMUNICAŢII

SCIENTIFIC BULLETIN

of

the „POLITEHNICA" University of Timişoara, Romania

Transactions on ELECTRONICS AND COMMUNICATIONS

Tomul 46(60), Fascicola 1, 2001
ISSN 1583-3380

EDITURA POLITEHNICA

Redactor şef / Editor in chief
Prof.dr.ing. Ioan Naforniță

Colegiul de redacție / Editorial Board:
Prof. dr. ing. Virgil Tiponuț
Prof. dr. ing. Alexandru Isar
Conf. dr. ing. Dan Lacrămă
Conf. dr. ing. Dorina Isar
Prof. dr. ing. Traian Jurcă
Conf. dr. ing. Aldo De Sabata
As. ing. Kovaci Maria - secretar de redacție

Colectivul de recenzare / Advisory Board:
Prof. dr. ing. Ioan Naforniță, UP Timişoara
Prof. dr. ing. Monica Borda, UT Cluj-Napoca
Prof. dr. ing. Brânduşa Pantelimon, UP Bucureşti
Prof. dr. ing. Ciochină Silviu, UP Bucureşti
Prof. dr. ing. Dumitru Stanomir, UP Bucureşti
Prof. dr. ing. Vladimir Crețu, UP Timişoara
Prof. dr. ing. Virgil Tiponuț, UP Timişoara

Buletinul Ştiinţific
al
Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ ŞI TELECOMUNICAŢII
TRANSACTIONS ON ELECTRONICS AND COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

SUMAR

ELECTRONICĂ APLICATĂ

Lucian A. Jurcă:
"Using Masked Logarithm Method for Fast Multiplication and Division Logarithmic Unit Design"... 3

Lucian A. Jurcă, Beniamin Drăgoi:
"Using 4:2 L Compressor in Partial Product Reduction Tree for Fast Binary Multipliers"... 11

Avram Ionel Adrian:
"Phycal Causes of Integrated Circuits Behaviour Degradation under Ionising Irradiation"... 17

INSTRUMENTAŢIE

Mihaela Lascu :
"Analog Input LabVIEW Applications for Measuring Temperature"... 19

Septimiu Mischie, Alimpie Ignea :
"Implementation of an AC Bridge using a Lab PC+ Data Acquisition Board"... ... ... .23

Daniel Belega, Septimiu Mischie :
"Considerations Concerning the Practical Behaviour of a Digitising Oscilloscope in Different Signal Recording Modes"... ... ... .27

TELECOMUNICAȚII

Muguraş D. Mocofan, Corneliu I. Toma:
"New Interactive Multimedia Application Model Using Objects Storage in a Database".…… 31

Ioan Şnep, Dan L. Lacrămă, Corneliu I. Toma:
"Optimised Homotropic Structuring Element for Handwriting Characteres Skeleton"... 37

Andrei Cubiţchi:
"Une méthode nouvelle pour la compression de la parole"... ..41

Corina Botoca:
"Romanian Bank-Notes Recognition Using Cellular Neural Networks"... .47

Konrad V. Pfaff:
"Evaluating Objective Audio Quality for Minidisc and Influence of Multiple Generations Audio Coding for Perceptual Quality"... 51

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Using Masked Logarithm Method for Fast Multiplication and Division Logarithmic Unit Design

Lucian A. Jurca[1]

Rezumat – În lucrare este prezentata o metodă originală de structurare a arhitecturii unei unităţi de calcul logaritmice, denumită metoda logaritmului mascat, ce permite efectuarea foarte rapidă a operaţiilor de înmulţire şi împărţire în simplă precizie. Viteza de calcul, superioară altor realizări prezentate în literatura de specialitate, se obţine în condiţiile în care aria ocupată pe cip şi erorile de calcul sunt mai mici decât în cazul referinţelor studiate.

Cuvinte cheie: unitate de calcul logaritmică, format LNS, format virgulă flotantă, simplă precizie.

I. INTRODUCTION

The circuits for computing logarithms and anti-logarithms allow the performing of the multiplication and division operations of two operands A and B by means of addition and subtraction operations:

$$A \times B = \text{antilog}(\log A + \log B) \quad (1)$$

$$A / B = \text{antilog}(\log A - \log B) \quad (2)$$

In fact, through performing a logarithmic and anti-logarithmic operation we accomplish the format conversion from floating-point format (FP) into logarithmic format (LNS – Logarithm Number System) and vice-versa.
Thus a binary number A in FP system, in single precision format is written:

$$A = (-1)^S (1 + 0.M)\, 2^{E-127} \quad (3)$$

where S represents the sign bit, M - the normalized mantissa, represented on 23 bits and E - the biased exponent, represented on 8 bits.
In the Logarithm Number System (LNS) a binary number z is represented:

$$z = (-1)^{Sz}\, 2^{Nz} \quad (4)$$

where S_Z is the sign bit and N_Z is a fixed-point number having n bits, out of which i bits for the integer part I_Z, and f bits for the fractional part F_Z. We have:

$$n = i + f \quad \text{and} \quad N_Z = I_Z + F_Z \quad (5)$$

The only deficiency of the LNS processors is that they so far permit only single precision computation and zero is not representable in the LNS. For system compatibility, a simple but slightly inefficient encoding is used. Zero is represented by a distinct bit Z in the LNS. When the bit Z is set to 1, the number is zero, regardless of all other bits in the value. Thus i=8 and f=23, which including the bit Z and the sign bit, corresponds to 33 bits [3].
Considering the normalized mantissa (1+0.M) in the domain [1,2), the "biased" integer part of the logarithm of the number is given by the very value of the biased exponent and the fractional part by the logarithm of the mantissa. This concatenation is possible because the binary logarithm of the normalized mantissa is always a positive fractional number.
The calculation of logarithm and antilogarithm use the partition of the argument and the memorising of only certain values for reducing the amount of memory and, in addition, we apply a correction method based on the memorisation in the same points of the values of the function derivative too, after which the interpolation is performed [1], [2]. Thus we note y = 0.M and we partition the mantissa y in two parts: y_1 containing the most significant 11 bits and y_2 containing the least significant 12 bits. The values of the function log(1+y)-y in these 2^{11} = 2048 points are memorised in internal ROM (ROMA) as correction values E_y provided through the application of the address y1.
Thus we obtain the following approximation:

$$\log(1+y) \cong y + E_y \pm \Delta E_y \times y_2 \quad (6)$$

The second look-up table (ROMA') of the memory is much smaller because $\Delta E_y << E_y$. Considering for ΔE_y a 12-bit representation, the complete conversion between the two formats is made through a reading in the look-up tables, two 23-bit additions and a 12×12 bit multiplication.

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Electronică Aplicată, Bd. V. Pârvan Timişoara 1900 e-mail jlucian@ee.utt.ro

The calculation of the antilogarithm is made in the same way. Considering C the result of finding the antilogarithm we can write:

$$C = 2^{E \pm 127 + 0.M} = 2^{E \pm 127}\, 2^{y} \quad (7)$$

where E represents the integer part in LNS format and M represents the fractional part. In this approach we do not extract the bias value from the exponents of the two operands. For this we extend the ALU with one bit to the left, while the bias value is extracted or added to the resulted exponent, depending on the performed operation, multiplication or division [4], [5]. Obviously, the implementation of the square root operation supposes the extraction of the bias value (just in this case) from one of the two data inputs.
Y is partitioned in the same way and we use a ROM (ROMC) for memorising the conversion error E_y in 2048 points, as well as the difference ΔE_y (ROMC'). The final result of conversion is:

$$2^{y} \cong (1+y) - E_y \pm \Delta E_y \times y_2 \quad (8)$$

The correction values E_y, for both log(1+y)-y and (1+y)-2^y are represented in figure 1.

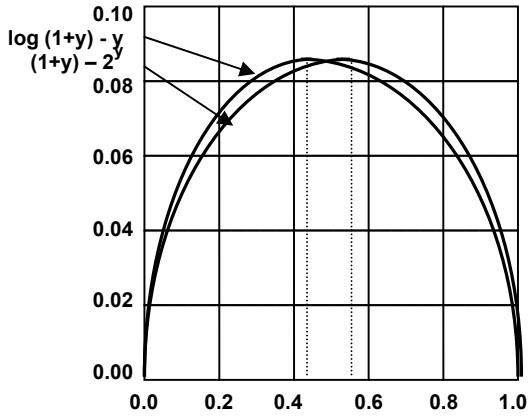


Fig.1 The conversion errors between the log(1+y) and y and the (1+y) and 2^y

Implementing the equations (1), (2), (6) and (8) led to a 6-stage pipeline structure [1], [2], which allowed a 100 MHz clock frequency, in 0.8 μm CMOS technology.
However the structure which was obtained has the following disadvantages from the point of view of the computation speed:
-The propagation times on the pipeline stages were not the same;
-The speed advantage resulted from the vertical carry propagation in the multiplication area, organised in a Wallace tree, was diminished by the time necessary for the proper addition of the last two intermediate partial products (when the carry propagation is horizontal), even if this is done with a fast adder;
-in the third stage of the structure, there are two successive proper additions, (which, together, lead to a unique final result), consuming a lot of time for the horizontal carry propagation;
-ALU operated with data of any polarity, which complicated its control logic and led to a further delay.
Later on, the same author presented a new architecture in which the product $E_y \times \Delta E_y$ was calculated not with binary multipliers but with PLA circuits [3], which permitted a saving of area on the chip, however maintaining the same computation speed.
Considering these disadvantages, this paper proposes a new method of logarithmic unit design, which does not give up the multipliers and leads to a substantial increase of the calculation speed of at least 1.6 times, resulting an occupied area which is smaller than in [1] and comparable to [3].

II. THE MASKED LOGARITHM METHOD

In order to eliminate the disadvantages mentioned above we offered a solution of our own, based on a new method of logarithmic unit design, which we called the "masked logarithm" method. To elaborate this method we considered the following furthers aims:
-the new structure should be organised in no more than 6 pipeline stages;
-the number of logical gates used on certain pipeline stages should not increase significantly;
-the size of the latches between the certain pipeline stages should not increase significantly;
-the propagation time on the critical stage of the pipeline should be as close as possible to the physically achievable limit, namely to the access time to the ROM integrated on the chip;
-the propagation times on the pipeline stages should be as close as possible.
Through this new approach, the terms $y_{A,}$, $Ey_{A,}$, respectively y_B, Ey_B which are added to the products $\Delta Ey_A \times y_{2A}$, respectively $\Delta Ey_B \times y_{2B}$ (see equation 6), will be considered as initial partial products of the above mentioned products. These terms will thus be introduced in the Wallace tree for the reduction of the number of partial products, beside the 12 initial partial products of each product. In this way, we eliminated the final multiplication adders, and the two resulting terms in the lower end of the trees (in the sum and carry sections), will implicitly contain the product value too, distributed between the two sections. Using this redundant representation, the proper values of the two products $\Delta E_y \times y_2$ will no longer be found in the structure explicitly, thus becoming "masked" values.
The problem which is still to be solved is that of the situation in which in equation (6), the term $\Delta E_y \times y_2$

is negative and its two's complement conversion, i.e. of all the 12 partial products, would be necessary. This happens starting from the address 907 to 2047 of ROMA and ROMA', situation which corresponds to the negative slope on the diagram of the function log(1+y)-y, presented in figure 1.

We managed to avoid this severe shortcoming through an artifice, which allows the total elimination of the cases in which the product $\Delta E_y \times y_2$ must be subtracted. As shown in figure 2, we can write the following equation:

$$Ey_{(n)} - \Delta Ey_{(n)} \times y_2 = Ey_{(n+1)} + \Delta Ey_{(n)} \times (\overline{y}_2 + 1) = Ey_{(n+1)} + \Delta Ey_{(n)} \times \overline{y}_2 + \Delta Ey_{(n)} \quad (9)$$

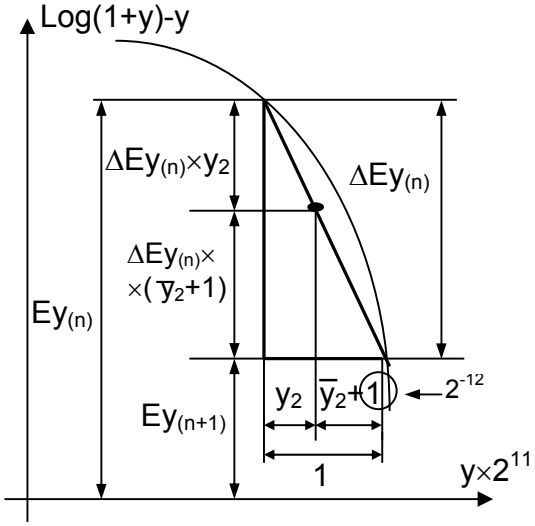


Fig.2 Negative slope segment achieved through linear interpolation between two consecutive memorised values E_y

The implementation of this equation leads to an arrangement of the partial products as they are presented in figure 3. A generic presentation, with "q_y" for the complemented "p_y" bits of y_2, respectively with "p_d" for the bits of ΔE_Y, was used.

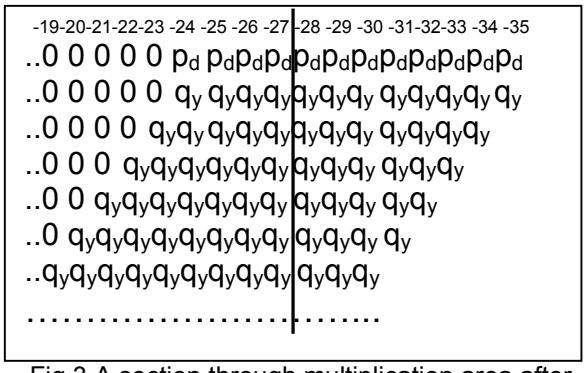

-19 -20 -21 -22 -23 -24 -25 -26 -27 -28 -29 -30 -31 -32 -33 -34 -35
..0 0 0 0 0 pd pd pd pd pd pd pd pd pd pd pd pd pd
..0 0 0 0 0 qy qy qy qy qy qy qy qy qy qy qy qy
..0 0 0 0 qy qy qy qy qy qy qy qy qy qy qy
..0 0 0 qy qy qy qy qy qy qy qy qy qy qy
..0 0 qy qy qy qy qy qy qy qy qy qy
..0 qy qy qy qy qy qy qy qy qy qy
..qy qy qy qy qy qy qy qy qy qy
.....

Fig.3 A section through multiplication area after the implementation of equation (9)

Thus, we can obtain the same result of the logarithm computation if we implement the right part of equation (9). Starting from the memory location corresponding to the address 907 of ROMA, instead of memorising the value $Ey_{(n)}$, $Ey_{(n+1)}$ is memorised, i.e. exactly what should have been found at the next address. In each location a supplementary bit will be memorised, called control bit, which takes the value 0 for addresses 0…906, and 1 for addresses 907…2047. If this bit is 1 then the generation of partial products will be done with y_2 having the bits reversed, and another pseudo-partial product with a size equal to that of the least significant partial product, having the value $\Delta Ey_{(n)}$, will be added. If the control bit is 0, then the generation of partial products will be done with y_2 unreversed, and the bits "p_d" of the first pseudo-partial product from figure 3 will all be 0.

The Wallace tree will thus have as inputs 15 initial pseudo-partial products and it will provide two data words: "sum" and "carry". If in this stage we did the proper addition of these, we would obtain the value of the logarithm of the mantissa of each operand applied to the input of the two logarithm computation circuits working in parallel. Further on, the two fractional numbers obtained would be concatenated to the exponents of the two operands, in order to obtain the logarithms of the operands. Finally, the two logarithms would be applied to the ALU, in order to be added or subtracted. However, in this case too, we would have two consecutive proper additions, which slows down the process of obtaining the result.

For avoiding this situation, we can also consider the two pairs of data words "sum" and "carry" as pseudo-partial products, and thus they are again introduced in a new reduction block. This will provide, in the end, two data words, the final "sum" and "carry", which will be possible to be added then, with the help of a fast adder. This final reduction block of the last pseudo-partial products will be included in the ALU as it should act under a control logic to allow the implementation of both addition and subtraction.

In this way, the adders which provide the explicit values of the logarithms of the mantissas for the two operands were practically eliminated. These values become hidden, being contained implicitly by the 4 partial pseudo-products, and they contribute to the process of obtaining the final result. That is why this operation method was called "the masked logarithm method".

The implementation of this method leads to the generating of a big Wallace tree which has 30 pseudo-partial products and which has its lower end in the ALU. We used, as reduction blocks for the pseudo-partial products, 4:2 compressor blocks and CSA blocks of 3:2 full adders, in such a way as to obtain the most efficient structure, in which there are no unoperated intermediary pseudo-partial products on any of the levels of the tree. The resulting structure is presented in figure 4.

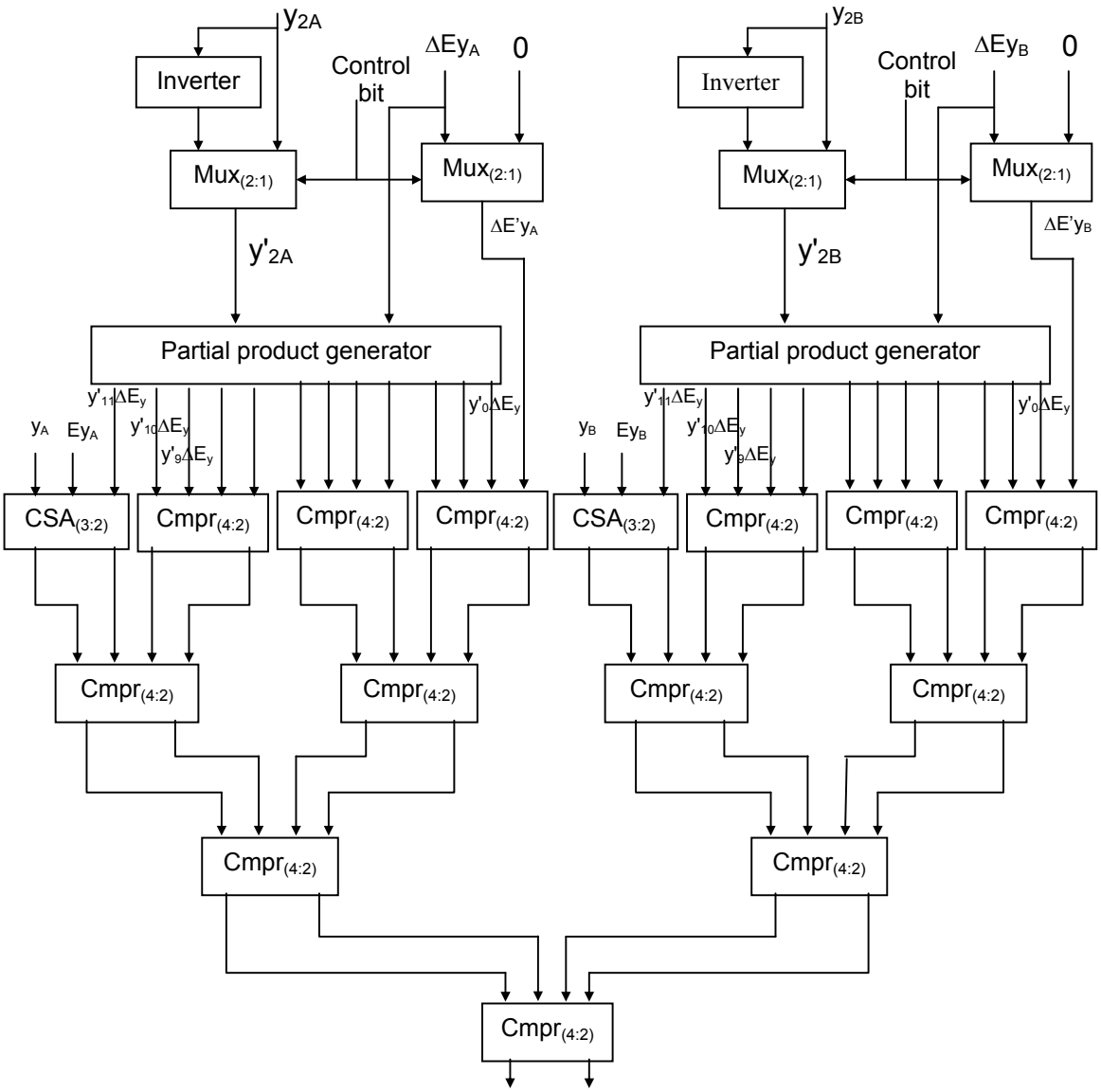


Fig.4 Wallace tree for pseudo-partial product reduction

The CSA block groups up the terms y and ΔE_y with the most significant partial product of $y_2 \times \Delta E_y$, in order to avoid the excessive growth of the size of this block, and so the size of the blocks connected after it as well as that of the wiring.

For the antilogarithm computation circuit, the procedure applied is the same, the data words "sum" and "carry" being obtained after only 3 levels of compressors. They are then added with a fast adder in order to obtain the mantissa of the final result from the output of the logarithmic unit. In this case too, we will take measures to avoid the subtraction of the term $\Delta Ey \times y_2$, but we also take into consideration the fact that in equation (8) the term Ey must be subtracted too. The product $\Delta Ey \times y_2$ is subtracted in the cases which correspond to the positive slope on the diagram of the function $1+y-2^y$, presented in figure 1, while it is added in the other cases. As equation (9) can no longer be used, the sum of the two negative terms from equation (8) will be written as follows:

$$-Ey_{(n)} - \Delta Ey_{(n)} \times y_2 = -(Ey_{(n)} + \Delta Ey_{(n)} \times y_2) = -[Ey_{(n)} + \Delta Ey_{(n)} - \Delta Ey_{(n)} \times (y_2+1)] = -(Ey_{(n)} + \Delta Ey_{(n)}) + \Delta Ey_{(n)} \times y_2 + \Delta Ey_{(n)} = -Ey_{(n+1)} + \Delta Ey_{(n)} \times y_2 + \Delta Ey_{(n)} \quad (10)$$

Thus, in the ROMC locations from the address 0 to the address 1082 the two's complement of the quantities $Ey_{(n+1)}$, which should have been found at the next address, as well as the value 1 for the control bit are memorised; from address 1083 to address 2047 (when $\Delta Ey \times y_2$ is positive) the two's complement of the values $Ey_{(n)}$, as well as the value 0 for the control bit will be memorised directly.

III. ALU DESIGN

Due to the fact that we must implement in the ALU, besides addition, the subtraction, too which implies the conversion into a two's complement of the subtractor (the number which is subtracted from the other term), the last 4:2 compressor block from figure 4 will definitely need to be included in the ALU. It will perform the reduction of the last four pseudo-partial products, after the control logic of the adder/subtracter circuit has intervened on the subtractor.

The subtractor is represented by two pseudo-partial products, whose addition is no longer performed, exactly with the purpose of increasing the propagation speed of the carry through the logarithmic unit. This means that both terms must be converted in two's complement code. The avoidance of the reconversion from two's complement in sign-magnitude code of the result, in the case in which it is negative, is done using the same method as in [6], only modified for 4 operands.

We note A_1, A_2, respectively B_1, B_2 the 4 pseudo-partial products, and $A = A_1 + A_2$ represents the subtractend, while $B = B_1 + B_2$ represents the subtractor, in the case in which a subtraction is performed. Considering the convention of representing B negated (the data word obtained through the inversion of all the bits of B) as being "/B", we can write the two terms which are simultaneously computed in the adder/subtracter circuit:

$$A - B = A - B_1 - B_2 = A + /B_1 + /B_2 + 2 \quad (11)$$

$$/(B - A) = A + /B_1 + /B_2 + 1 \quad (12)$$

Equation (12) can be checked as follows**:**
-in equation (13) below,

$$A - B = - (B - A) = /(B - A) + 1 \quad (13)$$

we replace the term"/(B-A)" with the value obtained from equation (12);
-we indeed obtain

$$A - B = (A + /B_1 + /B_2 + 1) + 1 = A + /B_1 + /B_2 + 2,$$

i.e. exactly the correct value from equation (11).

The principle of the adder/subtracter circuit is the following:
-if A-B is positive, it is selected at the output, and the sign bit of the result will be 0;
-if A-B is negative, it means that B-A is positive and, at the output, the term /(B-A) negated is selected, i.e. B-A; the sign bit of the result will be 1.

In order to apply the method from [6], the equations (11) and (12) are rewritten as follows:

$$A - B = (A_1 + A_2 + /B_1 + /B_2 + 1) + 1 \quad (14)$$

$$/(B - A) = (A_1 + A_2 + /B_1 + /B_2 + 1) + 0 \quad (15)$$

In other words, we maintain the situation of initial carry-in 1 , respectively that of initial carry-in 0 at the two adders which work in parallel, conditioned by the presence of a carry-in equal with 1 at the last 4:2 compressor block, in the case of performing a subtraction. Obviously, this carry-in will be 0 in the case of addition. The carry-in is applied at the input C_{in} which is not used of the least significant 4:2 compressor.

Further on, the length of the last block of the tree, included in the ALU, will be supplemented with 9 bits to the left, for the concatenation of the positive exponents (with the bias value of 127 included, according to the modified algorithm) which represent the biased integer part of the logarithm of the two operands. The fractional parts of the logarithms are "hidden", being distributed between A_1 and A_2, respectively B_1 and B_2 of the two pairs of pseudo-partial products. The concatenation of the exponents will be done at the terms A_1 and B_1, obtaining the final pseudo-partial products $\underline{A_1}$ and $\underline{B_1}$, while in the 9 bit positions of integer part corresponding to A_2 and B_2 of fractional weight, it will be completed with zeros. Thus we will obtain the other two final pseudo-partial products $\underline{A_2}$ and $\underline{B_2}$.

The block diagram of the new adder/subtracter, designed with the purpose of the complete implementation of the masked logarithm method, is presented in figure 5.

The bit line S_{OP} has the logical value 0 in the case of addition and 1 in the case of subtraction.

The three blocks "Invert." will be transparent when an addition is performed and they will invert the bits of the data from the inputs when a subtraction is performed. The block Selector will select the result from the Adder1 unchanged, in the case of addition, and in the case of subtraction it will select the result from Adder2 or from Adder1 inverted, depending on the MSB of Adder2 [5].

The Adder1 and Adder2 are 36-bit adders. The four least significant bits from the outputs of the two adders will be lost, and thus, at the output of the circuit, we will regain the single precision format plus one bit in the MSB position, which avoids the overflow due to the accumulation of bias values in the case of multiplication.

The justification of the 36-bit length for the adders and the compressor block will be done in the next paragraph.

The adders are 2-level hybrid adders, with four blocks of 8-b carry look-ahead adders (CLA) plus the most significant block of 4-b CLA in the 1st level and a carry select mechanism in the 2nd level.

In the most unfavourable case, the delay due to the carry propagation through the circuit presented in figure 5, was found 4 ns by simulation with MSim program, in 0.5μm CMOS technology.

The maximum propagation time through the Wallace tree which contains three levels of 4:2 L compressors is 3 × 1.1 ns, at which 0.7 ns are added for the generation of the partial products, which leads to a total of 4 ns.

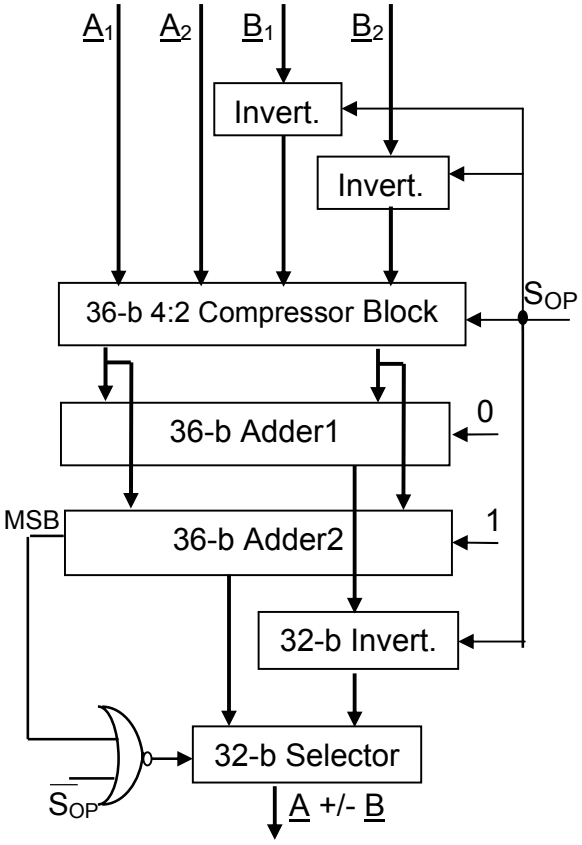


Fig.5 Block diagram of the ALU
(the adder/subtracter circuit)

The sign bits of the operands are processed separately in the ALU. The sign bit of the result can be obtained by the implementation of the logical function:

$$S_C = S_A \oplus S_B \qquad (16)$$

IV. ERROR ANALYSIS

After analysing, with the help of the Matlab program, the errors introduced by using the algorithm described in [1] for the generation of binary logarithm and antilogarithm, we had the confirmation of the value of 3×10^{-7} initially mentioned in [1] as the maximum conversion error. However, this error is of about 2.5 times bigger than the one in single precision format, which is of 1.19×10^{-7}. For the further minimisation of the conversion error, we suggested a solution of our own, which will add correction values on certain address intervals of the ROMA and ROMC, in the corresponding memory locations. Baring in mind that the error in floating-point single precision format, i.e. the value of the least significant bit provided by any output of the ROMA or ROMC, is 1.2×10^{-7}, it means that to the calculated values of E_y we can add corrections of one LSB, after which they are directly memorised (ROMA), respectively, they are transformed in two's complements and then memorised (ROMC).

Theoretically, the total conversion error can be reduced to half if to the memorised value E_y we add an amount equal with the half of the maximum conversion error, i.e. 1.5×10^{-7}. This amount can not be represented on an integer number of bits, which could have been 1 or 2. The compromise is thus the adding of 1 (one LSB) in those locations for which the conversion error is bigger than one LSB, i.e. bigger than 1.2×10^{-7}.

In this way the conversion error, both for logarithm and for antilogarithm computation, decreased to 1.8×10^{-7}.

As far as the computation of the product $\Delta Ey \times y_2$ is concerned, we can notice from figure 3, in which the multiplication area is presented, that, if we perform the calculation of the truncation error, in the most disadvantageous case, when all bits of a smaller or equal order with "-28" (the bits of the right side of the vertical line) are equal to 1, we obtain the value:

$$2\times10^{-35}+3\times10^{-34}+4\times10^{-33}+5\times10^{-32}+6\times10^{-31}+7\times10^{-30}+ +8\times10^{-29}+9\times10^{-28} = 5{,}96\times10^{-8}.$$

This value represents half of the representation error in single precision format. The removal of all bits from this area, i.e. the elimination of the hard structures from the whole Wallace tree and ALU which correspond to this area, does not introduce a supplementary error if the values E_y which are memorised in ROMA are rounded to 23 bits toward the nearest higher value, and the values E_y whose two's complement is memorised in ROMC are rounded through truncation.

V. CONCLUSIONS

The use of the masked logarithm method in the implementation of the logarithmic unit allows a very efficient organisation in a 6-stage pipeline structure, as shown in figure 6.

The total time for the obtaining of the result (we have taken into consideration the maximum propagation times of all the blocks from the critical path) is:

$$t_{tot} = t_{ROMA} + t_{log.tree} + t_{ALU} + t_{ROMC} + t_{alog.tree} + t_{FinSum} =$$
$$= 4 \text{ ns} + 4 \text{ ns} + 4 \text{ ns} + 4 \text{ ns} + 4 \text{ ns} + 4 \text{ ns} =$$
$$= 24 \text{ ns}.$$

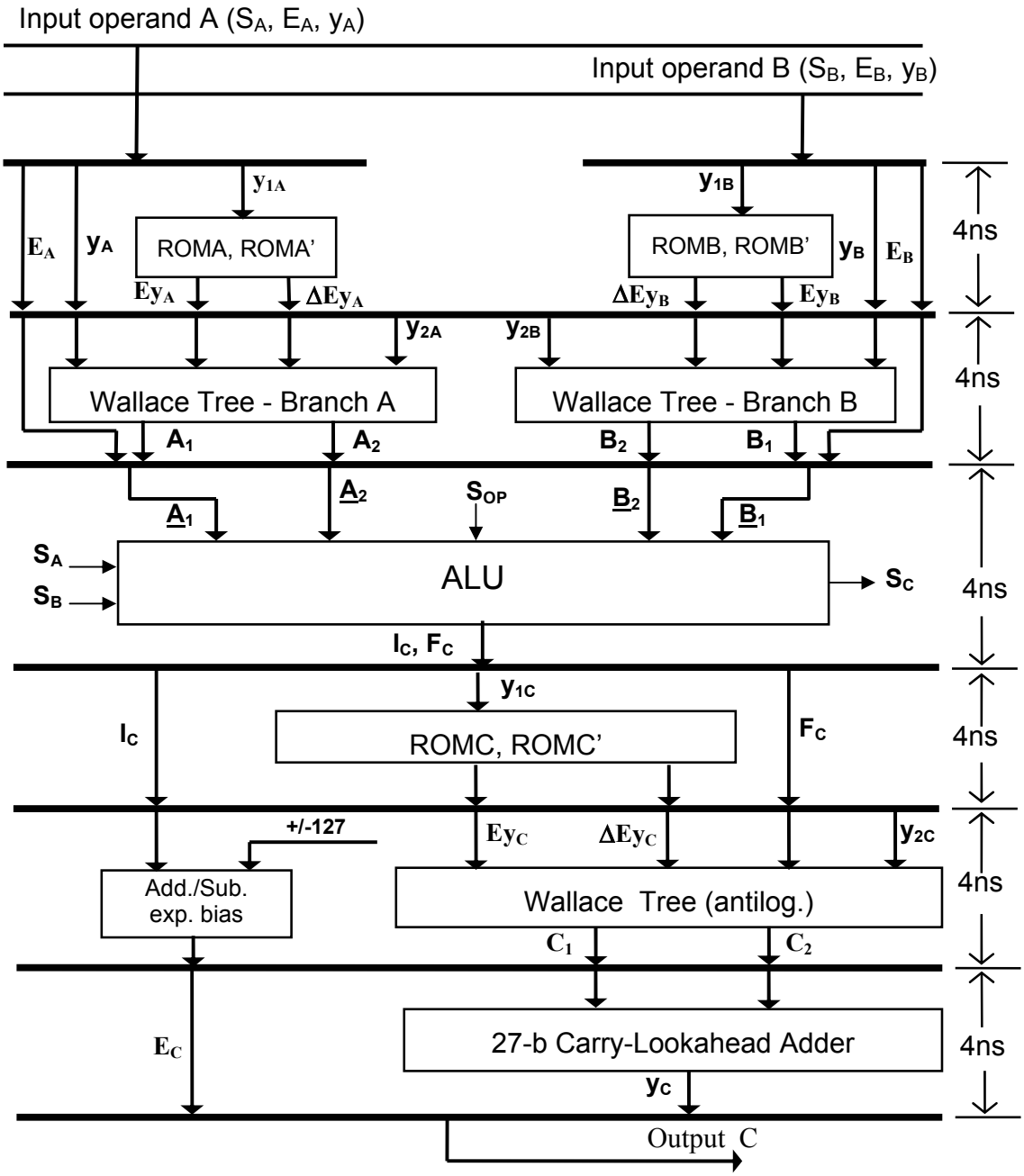


Fig. 6 The architecture of the logarithmic unit

We can see that in this case, unlike in the case of the most performant implementation described in the field literature by F. Lai and E. Wu [1], [2], [3], the propagation times through the blocks from the critical path are considerably smaller and are much more balanced.

The clock frequency of the system thus reaches 250 MHz, which represents a speed increase of over 60% in comparison with the above mentioned reference.

Furthermore, we can notice some other remarkable features:

-the number of logical gates from the entire structure is significantly smaller, due to the elimination of 8 fast adders of minimum 18 bits, with the price of the supplementation of three fast adders with 4 bits;

-the propagation time through the critical stage of the pipeline decreased up to the minimum physical level, given by the access time of the memories;

-the latch size between the pipeline stages decreased with roughly 15%, due to the arborescent structure of the logarithmic unit;

-a supplementary saving, of roughly 25%, was achieved at the implementation of the three multiplication operations of the type $\Delta Ey \times y_2$, a possibility which has been demonstrated through an error calculation in the case of the above mentioned product.

BIBLIOGRAPHY

[1] Lai F., "A 10-ns Hybrid Number System Data Execution Unit for Digital Signal Processing Systems", IEEE Journal of Solid-State Circuits, Vol. 26, No. 4, Apr. 1991, pp. 590-599

[2] Lai F., Wu C.F.E., "A Hybrid Number System Processor with Geometric and Complex Arithmetic Capabilities", IEEE Transactions on Computers Vol. 40 No.8 Aug.1991, pp. 952-961.

[3] Lai F., "The Efficient Implementation and Analysis of a Hybrid Number System Processor", IEEE Transactions on Circuits and Systems, Vol. 46, No. 6 ICSPE5, June 1993, pp 382-392.

[4] Jurca L. A., "Some Considerations Regarding the Design of a Hybrid Logarithmic, Floating-Point Mathematical Processor", Buletinul UPT, Tom 45(59), Fasc. 1, 2000, pp. 169-172

[5] Jurca L. A., "Logarithmic Unit for Fast Performing of Multiplication and Division Operations in Single Precision Format", Buletinul UPT, Tom 45(59), Fasc. 2, 2000, pp.29-32.

[6] Fujii H. & all, "A Floating-Point Cell Library and a 100-MFLOPS Image Signal Processor", IEEE Journal of Solide-State Circuits, Vol.27, No.7. July 1992, pp.1080-1088.

[7] Mori J. & all "A 10-ns 54×54-b Parallel Structured Full Array Multiplier with 0.5-μm CMOS Tehnology", IEEE Journal of Solid-State Circuits, Vol.26, No.4, April 1991, pp. 600-605.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Using 4:2 L Compressor in Partial Product Reduction Tree for Fast Binary Multipliers

Lucian A. Jurca[1], Beniamin Drăgoi[2]

Rezumat – În lucrare este prezentat un nou tip de compresor 4:2, denumit compresor 4:2 L ce permite o creştere cu aproximativ 10% a vitezei de calcul a unui multiplicator binar la care produsele parţiale sunt organizate în arbore Wallace. Sinteza compresorului 4:2 L a avut la bază o nouă abordare în definirea acestui tip de circuit, ce a condus la lărgirea familiei din care face parte. Simulările efectuate au dovedit avantajele în ce priveşte viteza de propagare a transportului prin noua structură, realizată în tehnologie CMOS 0,25 micrometri, comparativ cu alte soluţii prezentate în literatura de specialitate.

Cuvinte cheie: compresor 4:2, format LNS, format virgulă flotantă, simplă precizie

I. INTRODUCTION

The increased level of integration brought by modern VLSI and ULSI technology has rendered possible the integration of many components that were considered complex and were implemented only partially or not at all in the past. The multiplication operation is certainly present in many parts of a floating-point (FP) processor, most notably in signal processing, graphics and scientific computation. Parallel multipliers have even migrated into the fixed-point processor of digital computers for de purpose of speeding up and facilitating address calculation and, more recently, into the hybrid FP-LNS (Logarithm Number System) processors [8], [9], for de purpose of allowing implementation of algorithms needed for format conversion between FP and LNS.

The most appropriate multiplier is obtained when the partial products from the multiplication array are grouped into a Wallace tree [1]. Initially, the component blocks of the tree were CSA (Carry Save Addition) blocks, which were made up of independent 1-bit full adders. The CSA blocks accepted 3 partial products at the input and they provided 2 intermediate partial products in a time equal to the one necessary for the propagation through one full adder.

The great computation speed is obtained on the one hand due to the solely vertical propagation of the carry in the multiplication area, and on the other hand due to the simultaneous calculation of more initial or intermediate partial products.

But using only CSA blocks leads to the situation in which 1 or 2 partial products remain unoperated on one of the levels of the Wallace tree. This fact results in a certain inefficiency due to the further delay which appears with the increase of the number of levels of the Wallace tree. A more efficient structure is obtained if, instead of the CSA blocks, we use 4:2 compressor blocks or a combination of 4:2 compressors and CSA blocks.

The 4:2 compressor was introduced in the field literature by Weinberger in 1981. However, the first implementation on the chip which used this circuit appeared only in 1991, in the form of the 54×54 bit multiplier, designed by J. Mori [2], possible to be achieved as a result of the increase of integration density. Since then, other, improved variants of 4:2 compressors have appeared, variants which will be compared at the end of this paragraph with our own variant of this circuit, named the 4:2 L compressor.

II. THE 4:2 COMPRESSOR CIRCUIT

The generalised name of “4:2 compressor” does not reflect exactly the features of this circuit because the sum of the 4 input bits of equal weight can not be represented in all possible cases with the help of the two output bits. The circuit also provides a carry-out on a supplementary output line and thus needs an input for the introduction of the carry-in. In this way, there is a “4:2” vertical propagation path and a “1:1” horizontal propagation path. However, the horizontal propagation of the carry is practically limited to 1 bit, because the carry-out is generated in such a way that it does not depend on the carry-in. Thus, on the scale of the whole multiplier, the use of this circuit permits the shift of the critical propagation

[1, 2] Facultatea de Electronică şi Telecomunicaţii, Departamentul Electronică Aplicată, Bd. V. Pârvan Timişoara 1900
[1] e-mail: jlucian@ee.utt.ro [2] e-mail: bdragoi@yahoo.com

path from the centre of the multiplication area (where the degree of partial product overlap is the maximum) towards the left end (where this degree of overlap becomes less and less).
Furthermore, the combination of CSA blocks of 3:2 full adders and 4:2 compressor blocks allows an efficient design of the Wallace tree, by the elimination of the cases in which one or two initial or intermediate partial products remained ungrouped cross a level of the tree without being operated [5]. The number of levels of the Wallace tree thus becomes smaller and the critical propagation path is also shorter.
In principle, a 4:2 compressor is obtained through the interconnection of two 3:2 full adders, whose structure is presented in figure 1 doubly and which presents a propagation time equivalent to 4 OR-exclusive gates in cascaded configuration.
A major improvement of the structure of the 4:2 compressor was brought by D. Villeger in [3] and [6], by the exploiting of the "fast" inputs and outputs of the 3:2 full adders, in such a way that the maximum propagation time becomes the equivalent of only 3 OR-exclusive gates in cascaded configuration. This optimised structure is presented in figure 1.
For this compressor, the 4 input bits are x1, x2, x3 and x4, while the 2 output bits are S_{um} and C_{arry}. The carry-in is C_{in} and the carry-out is C_{out}. The C_{arry} and C_{out} bits have a double weight in comparison with S_{um}. The Sum bit is obtained through the implementation of the OR-exclusive function between the 5 inputs of the circuit. Supposing that the propagation time of the carry through an OR-exclusive gate is twice bigger than through an NAND or NOR gate, we can notice in figure 1 that, indifferent of the propagation path of the input signals, including the C_{in} generated by the right-hand compressor, the maximum propagation time is not longer than the equivalent of 6 simple logical gates or 3 OR-exclusive gates.
We can also notice that the output C_{out} does not depend on the input C_{in}, and so the propagation of the carry on the horizontal in the compressor block from figure 2 is limited to a single bit (single position) to the left. The vertical propagation time of the whole block is actually the vertical propagation time through one compressor. This obviously means that the reduction to half of the number of partial products is made in a time equal to the propagation time on the critical path through the 4:2 compressor. For this reason, the fast propagation of signals through this circuit is essential and practically determines the propagation speed through the whole Wallace tree.
The optimised 4:2 compressor, described in [6] and presented in figure 1, has a series of limitations which affect the propagation time of the carry:

-The 3-input NAND gates, in advanced submicronic technologies which need smaller supply voltages, can no longer be implemented, or they are implemented through two 2-input NAND gates, by the addition of one more logical level. In this way the critical propagation path through the circuit becomes the one which includes the fast output C_{out} of the current compressor and the input C_{in} and the output C_{arry} of the left-hand compressor. So, this path becomes slower compared to the propagation time for "x1" and "x2" (figure 1) through 3 successive OR-exclusive gates. Even in technologies which allow the integration of 3-input NAND gates, its propagation time is bigger than through a 2-input gate.
-The use of unconventional OR-exclusive gates [2], faster than the classical ones, having $t_{P(OR\text{-}excl)} < 2t_{P(NAND)}$, will not bring the expected advantages because the propagation through 3 OR-exclusive gates in cascaded configuration no longer represents a critical path in the compressor.
-There are lots of cases in which the capacitive load of a logical output is considered to be excessive when it appears on a critical propagation path. Thus, there are cases in the structure of the compressor in which an output commands three inputs, i.e. minimum six MOS gates. To the intrinsic capacity of these, there is added the parasite capacity of the three signal paths.

III. THE 4:2 L COMPRESSOR

The elimination of the above mentioned disadvantages was made by the design of our own variant of 4:2 compressor, presented in figure 3, which is 10% faster than the compressor presented in [6], as proved by simulation. Considering the fact that in a 54 $\times$ 54-bit multiplication array there are 5 levels of 4:2 compressors, this leads on the whole to a saving of 0.35 ns (in 0.25 μm CMOS technology).
In the design of our own variant we started from the idea of finding a general definition for the 4:2 compressor, whose inputs and outputs are already defined. Thus, this circuit has to provide the following functions:
a) the output S_{um} has to be generated by the equation (1):

$$S_{um} = x1 \oplus x2 \oplus x3 \oplus x4 \oplus C_{in} \quad (1)$$

b) the two outputs C_{arry} and C_{out} have to be of an immediately superior order to S_{um} (to have a double weight), so that the maximum sum of the 5 input bits, which equals 5, could be represented. In this way, the equation (2) will be satisfied:

$$x1+x2+x3+x4+C_{in} = C_{arry}+ C_{out}+ S_{um} \quad (2)$$

Indeed, when all the input bits are 1 we obtain:

$5=10_2+10_2+1$.

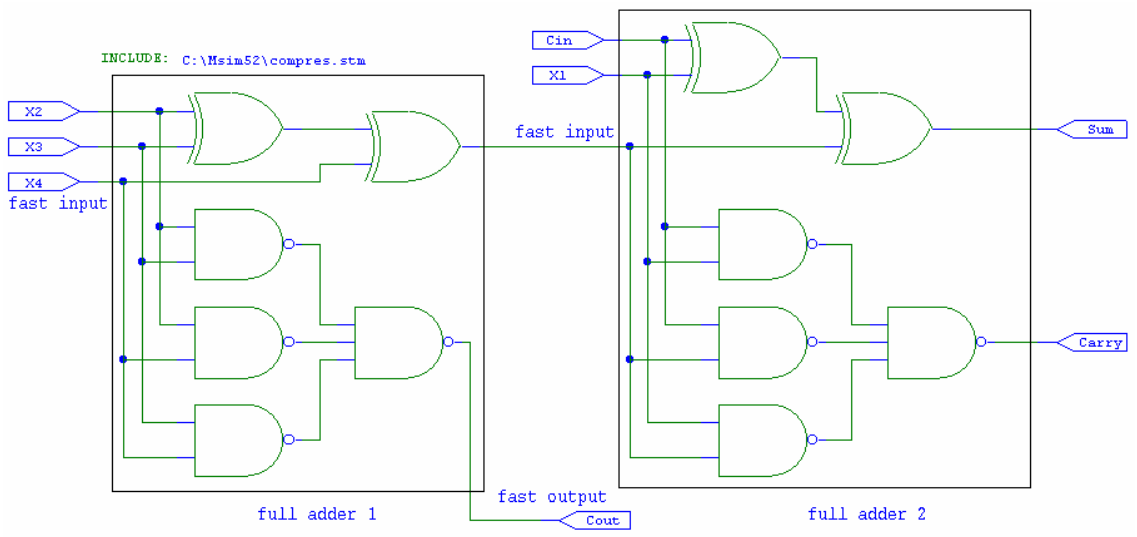


Fig. 1 Optimized 4:2 compressor, presented in [6]

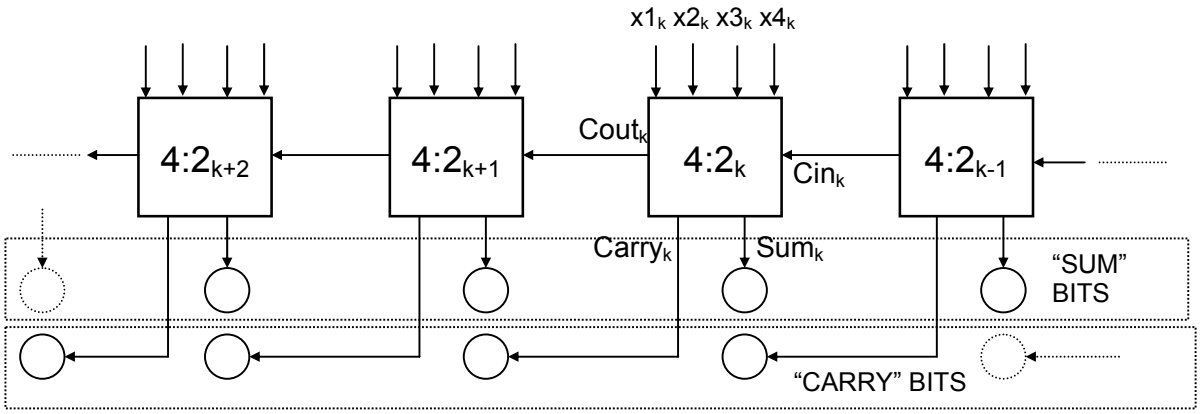


Fig. 2 Section through a 4:2 compressor block

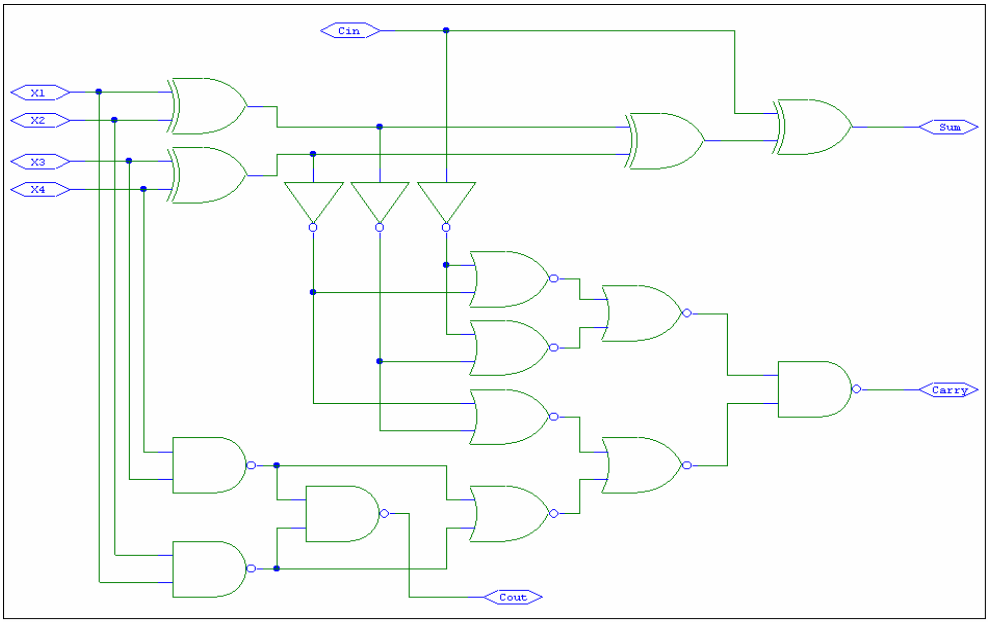


Fig. 3 The 4:2 L compressor

Because the outputs C_{in} and C_{arry} have the same order, they provide a redundant representation at the output of the 4:2 compressor.
c) One of the bit lines of output carry has to be generated by simple logical functions, which to include a minimal number of logical levels and not to depend on the carry-in C_{in}. It will be called fast output and will insure the carry-in for the compressor with the immediately superior order. In this way the horizontal propagation of the carry will be limited to only one bit and will not influence the speed of result generation at the output of a cascaded 4:2 compressor block.
d) All possible combinations of input bits for which their sum is 0 or 1 have to lead to $C_{arry} = C_{out} = 0$.
e) All possible combinations of input bits for which their sum is 2 or 3 have to lead to $C_{arry}= 1$ and $C_{out} = 0$, or $C_{arry}= 0$ and $C_{out} = 1$. In other words, all these possible combinations which produce either $C_{arry}= 1$ or $C_{out} = 1$ have to be found in two disjuncted areas. This possibility insures the redundant character of the outputs of the circuit. Exploiting of this property, we will be able to enlarge the family of 4:2 compressors.
f) All possible combinations of input bits for which their sum is 4 or 5 have to lead to $C_{arry} = C_{out} = 1$.
Taking into consideration all these functions, considered as compulsory, but also the disadvantages of the 4:2 compressor of reference, it resulted the efficient structure of the 4:2 compressor from figure 3, called the 4:2 L compressor.
Due to the big number of inputs for the generation of the S_{um} bit according to equation (1), we adopted an arborescent structure of OR-exclusive gates, which groups up the inputs x1, x2, x3, x4 and afterwards C_{in}. Then, based on the grouping of the first 4 inputs, we implemented C_{out} through the simple logical function:

$$C_{out}= x1\ x2 + x3\ x4 \qquad (3)$$

We thus obtain the generation of the fast output "C_{out}" by the crossing of only two logical levels, obtained only with intrinsic gates with only two inputs, applying the De Morgan equations. None of the input lines is loaded with more than two logical inputs anymore. We can notice that equation (3) fulfils the demands d) and f). In order to obtain the output "C_{arry}" we implemented equation (4):

$$C_{arry}= (x1 \oplus x2)(x3 \oplus x4)+C_{in}(x1\oplus x2)+C_{in}(x3 \oplus x4)$$
$$+x1\ x2\ x3\ x4 \qquad (4)$$

The first three terms of the logical sum insures the fulfilment of the demand e) related to C_{out}, while the fourth term insures the fulfilment of the demand f) from the above list. We can also notice that the demand d) is fulfilled by all four terms.
Using the three inverters from the figure 3 permitted the complete elimination of the disadvantage of supplementary capacitive load of certain logical outputs (one output does not command more than two inputs) and, in the same time, it permitted the implementation of the De Morgan equations for the exclusive use of intrinsic gates in CMOS technology.

IV. SIMULATIONS RESULTS. CONCLUSIONS

The circuits presented in figure 1 and figure 3 was simulated for all possible logic combination of the inputs. The logic state diagram for the outputs S_{um}, C_{arry} and C_{out} for the 4:2 compressor described in [6] is presented in figure 4 and for the 4:2 L compressor in figure 5. For each input combination the analysis time was initially established at 10 ns i.e. much greater than the propagation time through the compressor.
As we can see from figure 4 and figure 5, the output S_{um} is the same, according to the function a) in our definition for the 4:2 compressor. The outputs C_{arry} and C_{out} present the same logic state in both figures if the sum $x1+x2+x3+x4+C_{in}$ is equal with 0, or 1, or 5 according to functions d) and f). If the above sum is equal with 2 or 3, than $C_{arry}+ C_{out}$ is always 1 in both diagrams, according to the function e) of a compressor.
An analysis in terms of speed is made with the help of an analog simulator upon the layout of the circuits, using MOSIS models for 0.25 μm TSMC process. For evaluating the real propagation speed of the carry in the most unfavourable case, for both circuits, the input signals x3 and x4 are applied like x2 in figure 6 and figure 7, and C_{in} is applied after a delay which was found for both circuits, by simulation too. As we can see from the diagrams presented in these figures, the signal S_{um} is obtained faster in the case of the 4:2 L compressor than in the other case. The behaviour of the two circuits is obviously different, and the most unfavourable case is different for each compressor. For the 4:2 L compressor the maximum propagation delay was found 0.53 ns, while for the other circuit this delay was found 0.6 ns.
A comparison in terms of speed between our own variant of 4:2 compressor and several other recent variants that were published in the field literature, in 0.5 μm and 0.25 μm CMOS technology, is illustrated in the table I:

Table I.

Compres. type	[2] '91	[4] '95	[6] '96	[7] '99	4:2 L 2001
Tech.[μm]	0.5	0.5	0.25	0.25	0.25
Delay[ns]	1.4	1.4	0.6	0.6	0.53

As we can notice, the 4:2 L compressor is superior in speed to the other variants, even if they are implemented in the same submicronic technology.

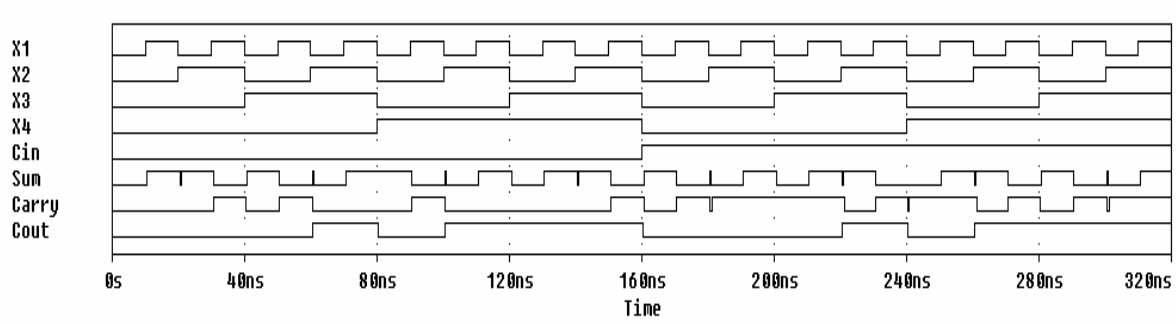


Fig. 4 The logic state of the outputs S_{um}, C_{arry} and C_{out} for all possible logic combination of the inputs, in the case of the 4:2 compressor presented in [6]

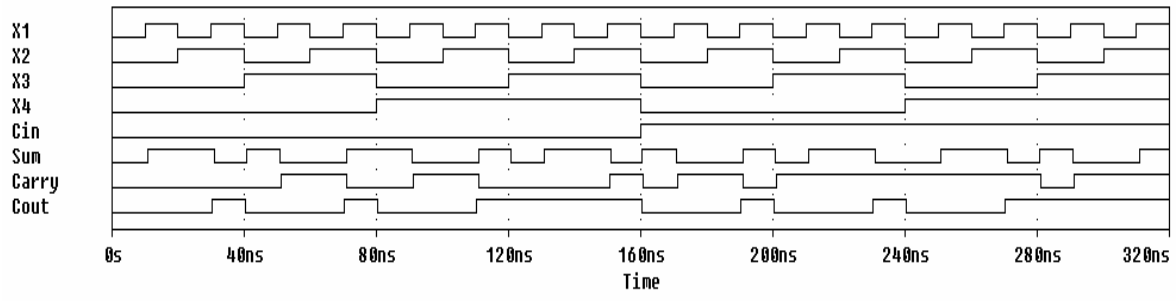


Fig. 5 The logic state of the outputs S_{um}, C_{arry} and C_{out} for all possible logic combination of the inputs, in the case of the 4:2 L compressor

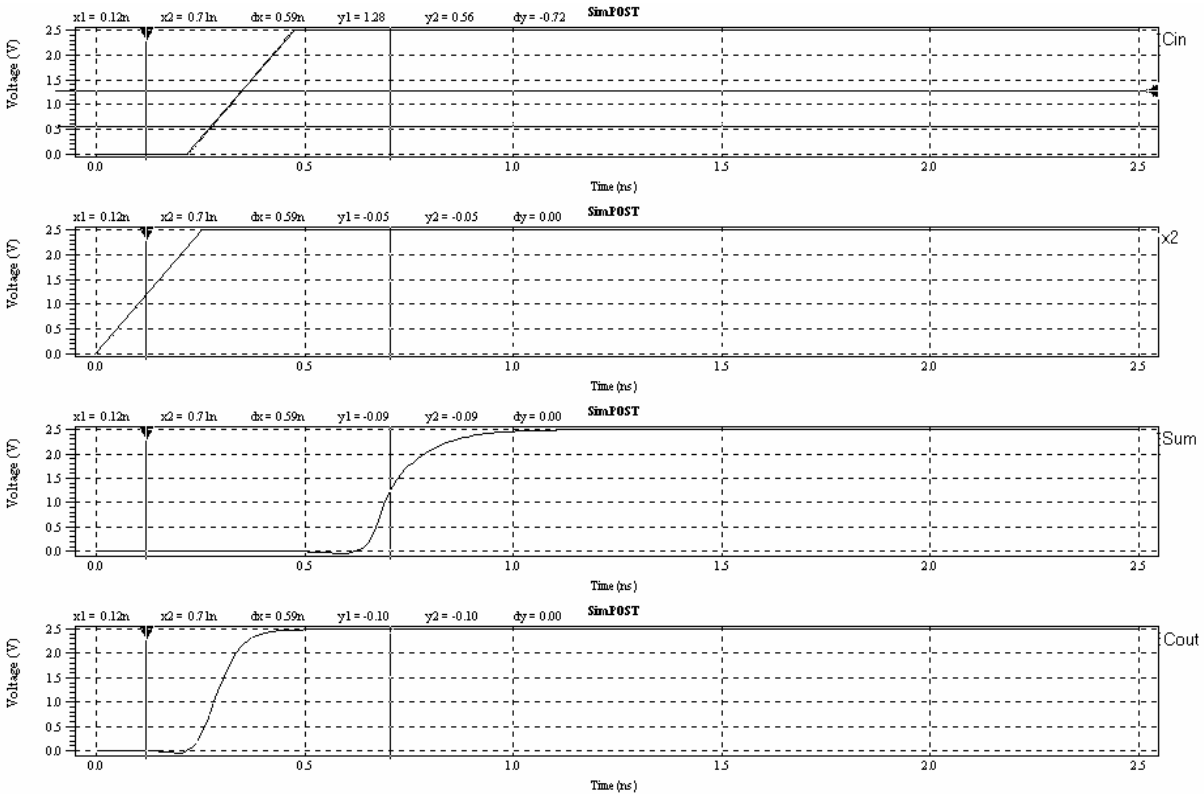


Fig. 6 The logic state of the outputs S_{um} and C_{out} in the most unfavourable case, in the case of the 4:2 compressor presented in [6]

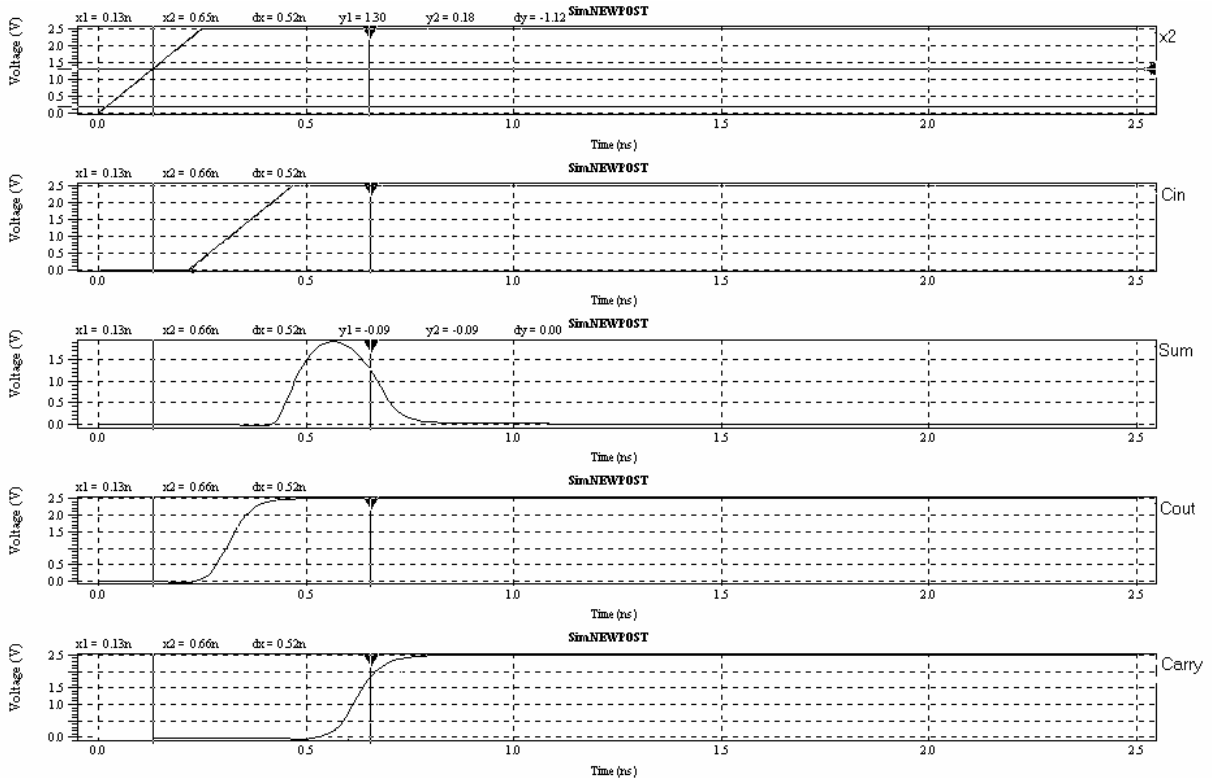


Fig. 7 The logic state of the outputs S_{um}, C_{arry} and C_{out} in the most unfavourable case, in the case of the 4:2 L compressor

BIBLIOGRAPHY

[1] Stallings W., Computer Architecture and Organization, Prentice Hall Inc, 1996.

[2] Mori J. & all "A 10-ns 54×54-b Parallel Structured Full Array Multiplier with 0.5-μm CMOS Tehnology", IEEE Journal of Solid-State Circuits, Vol.26, No.4, April 1991, pp. 600-605.

[3] Villeger D., Fast Parallel Multipliers, Final Report, Ecole Superieure d'Ingenieurs en Electrotechnique et Electronique, Noisy-le-Grand, May 1993.

[4] Ohkubo N. & all, A 4.4 ns CMOS 54×54-b Multiplier Using Pass-Transistor Multiplexer, IEEE Journal of Solid-State Circuits, Vol. 30, No. 3, March 1995, pp. 251-257.

[5] Wang Z., Jullien H. A., Miller W. C., A New Design Tehnique for Column Compression Multipliers, IEEE Transactions on Computers, Vol.44, No.8, August 1995.

[6] Oklobdzija V. G., Villeger D., Liu S. S., A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach, IEEE Transaction on Computers, Vol.45, No.3, March 1996, pp. 294-305

[7] Yan H. & all, A Low-Power 16×16-b Parallel Multiplier Utilizing Pass-Transistor Logic, IEEE Journal of Solid-State Circuits, Vol. 34, No. 10, October 1999, pp. 1395-1399.

[8] Lai F., "A 10-ns Hybrid Number System Data Execution Unit for Digital Signal Processing Systems", IEEE Journal of Solid-State Circuits, Vol. 26, No. 4, Apr. 1991, pp. 590-599

[9] Jurca L. A., "Logarithmic Unit for Fast Performing of Multiplication and Division Operations in Single Precision Format", Buletinul UPT, Tom 45(59), Fasc. 2, 2000, pp. 29-32.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Physical Causes of Integrated Circuits Behaviour Degradation under Ionising Irradiation

Avram Ionel Adrian[1]

Resume:

Modern research in integrated circuits (IC) fiability domain, show that varied IC fabrication technologies, schemotechnics and constructive particularity and exploitation condition leads to different behaviour of IC under ionising conditions.

Introduction:

For a precise design of IC under ionising irradiation conditions are necessary detailed information not only about irradiant physics process in semiconductor materials, but also are required information above IC characteristics modification under ionising irradiation and information about influence of IC structure organisation.

Difficulty of this problem is determinate by using different technologies, high integrated degree, functional complexity and by high standard of necessary levels to assure IC stability under ionising irradiation action [1].

To assure electronic devices fiability that works at various type of irradiation like: Röntgen, γ quanta, electrons, protons, neutrins and other natural and artificial type source, is an actual problem, because these types of irradiation leads to defects appearence in semiconductor devices that may by the cause of IC fiability reduction [1,2].

Variety of effects that appear in IC at ionising irradiation are determinate by many factors that exist in irradiant environment. Space is a real irradiant environment were exist: galactic irradiation - protons flow, α particles, etc. with energy between 10^2-10^{14} MeV; solar irradiation - α particles and protons flow with energy below galactic energy but with high flow density; Earth irradiation due Earth magnetic field - protons flow with energy low to 700MeV and electrons flow with energy low to 1MeV [1-3]. Artificial irradiation source may be made using reactors and other ionising irradiation equipment [1-4].

Discution:

Principal causes of IC, based on TECMOS and bipolar transistors, functional characteristics degradation under ionising irradiation action, are determinate by charges accumulation in oxide volume and substrate and at oxide-semiconductor interface (TECMOS case). These charges can be classified such:

1) Interface trapped charges, Q_{it}, localised at Si-SiO_2 interface. This type of charges induced energetic state in Si forbidden band;
2) Fixed oxide charges, Q_f, localised near Si-SiO_2 interface. This type of charges are imobile in electric field;
3) Oxide trapped charges, Q_{ot}, localised near Si-SiO_2 interface, or near oxide-metal interface. Oxide trapped charges are created by exposition at ionising irradiation;
4) Mobile ionic charges, Q_m, determinate by presence of alcaline metal ionised atoms such sodium and potassium. This type of charges are localised near Si-SiO_2 interface, and gets there by moving in electric field.

Using gate dependence, last three categories can be grouped together in one unique class “oxide charges, Q_{ot}”, different as interface trapped charges that depend by applied tension on gate an will be call “interface charges, Q_{it}”.

The effect of positive charge accumulation in MOS structures oxide, under ionising irradiation action was study by Grove A. S. , Snow E.I. [6] and Stanley A. G. [7]. In concordance with these models, under ionising irradiation action, in MOS structures oxide take place electron-hole pair generation followed by separation of these pair under external electrical field, according with passing of most mobile carriers - electrons - from

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Electronică Aplicată Bd. V. Pârvan Timişoara 1900 e-mail adiavram@hotmail.com

oxide in electrodes. At the same time take place less mobile carriers capture - holes - in oxide by trapped centres that are formed in broken bonds process between Si and O_2 atoms. Trapped holes process continue until take place the compensation of external electric field intensity with induced irradiation field of charges in Q_{ot} volume and ending of electron-hole pair divide process. These models permit a qualitative explication of saturation effect of $Q_{ot}(D)$ (that was experimental observable) in dependence of dose, were D - absorption dose. Also these models permit to understand the growing effect of Q_{ot} at saturation in condition of positive potential V_G growing on metalic electrode of irradiated MOS structure [1].

Holes that not be trapped, slowly begin to move to negative electrode. Surface state forming process take place in two phase: in phase one holes are formed that will be capture by "traps" and in second phase take place positive ions movement to Si-SiO_2 interface. After second phase interface state are formed. The problem of forming interface state have a principial contribution because interface forming process take place after ending irradiation process and after forming Q_{ot} charge [5].

Electrons and holes are separated in oxide by electric field, and positive charge Q_{ot} appear in oxide volume with N_{ot} density and interface state charge Q_{it} with N_{it} density. The sign of interface state charges is negative for TECMOS with n-channel and positive for TECMOS with p-channel.

Q_{ot} and Q_{it} charges sign have an essential value: charges growing in oxide volume contribute, for nTECMOS, to threshold voltage reduce, in a same time interface state charge growing increase threshold voltage. Variation of nTECMOS threshold voltage are given by:

$$\Delta V_{th} = (\Delta Q_{ot} - \Delta Q_{it})/C_0 \quad (1)$$

were: ΔQ_{ot} and ΔQ_{it} - charge modification in volume and on interface state; C_0 - specified gate-substrate capacity.

In TECMOS with p-cannel, both charges are positive. This thing leads to threshold voltage increasing. Threshold voltage variation of a pTECMOS is given by:

$$\Delta V_{th} = (\Delta Q_{ot} + \Delta Q_{it})/C_0 \quad (2)$$

Charges accumulation leads to punctual defects apparition in semiconductor structures determinate by chemical bonds broken. Ionising conditions leads to accumulation of punctual defects under structural defects. In long term working under ionising condition, charges are all times in bulk state. This type of state leads to appearence of IC disfunctionality.

One way to prevent the appearence of this disfunctionality is to "train" integrated circuits. "Training" consist in alternation of irradiation period with nonirradiation period. This treatment is based on fact that in nonirradiant period IC tends to return at initial state through charges annealing, namely through charge carriers recombination. This method is called "itself annealing". In such a way it may prolong life time of integrated circuits in irradiant conditions.

Conclusions:

IC stability at irradiation is determinate, in the first time, on transistor parameters degradation as much in bipolar structures and MOS structures. Ionising irradiation influence analysis above TECMOS and BJT active elements show that ionising irradiation through charge accumulation determinate in MOS structures significant degradation of follow parameters: threshold voltage V_{th}, mutual conductance (slope) g_m, carriers life time< for bipolar structures ionising irradiation determinate increasing of base current, I_B and decreasing of amplified coefficient h_{21E} determinate by increasing of speed recombination and structural defects called "clusters" [1].

References:

[1]. Avram Adrian, "Simularea şi prognozarea funcționării elementelor şi circuitelor integrate sub acțiunea iradierii ionizante", Teză de doctor, Chişinău, 2000.

[2]. R. M. Barnett et al., Review of particle properties, Phzs. Lett. D54 (1996).

[3]. Вавилов В· . С. "Действие излучений на полупроводники", М.:Физматгиз, 1963 - 263с.

[4]. Setz F. "On the disordering of Solids bz the Action of Fast Particles" - Discussion of the Faradaz Society. London, 1949, vol.5, p.271-282.

[5]. Schleiser K. M., Shaw J. M., Benyon J. M., "Al_2O_3 as a Radiation - Tolerant CMOS Dielectric" // RCA Rev. 1976, vol.37, No.3, p.350-388.

[6]. Grove A. S., Snow E. N., "A model for iradiation damage în metal-oxide-semiconductor", Proc. IEEE, 1966, V. 54, p.894-895.

[7] Stanlez A. G. "A model for shifts in the gate turn-on voltage of insulated-gate field effect devices induced by ionizing irradiation" IEEE Trans. Electron Devices, 1967, V.4, p.134-138.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Analog Input LabVIEW Applications for Measuring Temperature

Mihaela Lascu[1]

Abstract – Signal Conditioning eXtensions for Instrumentation (SCXI) is a highly expandable signal conditioning system. This paper describes the basic concepts of signal conditioning, the setup procedure for SCXI hardware, the hardware operating modes, the procedure for software installation and configuration, the special programming considerations for SCXI in LabVIEW, and some common SCXI and acquisition applications.

Keywords: SCXI, transducer, thermocouple, temperature, data acquisition.

I. INTRODUCTION

Transducers can generate electrical signals to measure physical phenomena, such as temperature, force, sound, or light. To measure signals from transducers, you must convert them into a form that a DAQ device can accept. For example, the output voltage of most thermocouples is very small and susceptible to noise. Therefore, you may need to amplify and/or filter the thermocouple output before digitizing it. The manipulation of signals to prepare them for digitizing is called signal conditioning. Common types of signal conditioning include the following: amplification, linearization, transducer excitation, isolation and filtering.

II. SCXI OPERATING MODES

The SCXI operating mode determines the way that DAQ devices access signals. There are two basic operating modes for SCXI modules, **multiplexed** and **parallel**. When an analog input module operates in multiplexed mode, all of its input channels are multiplexed to one module output. When you cable a DAQ device to a multiplexed analog input module, the DAQ device has access to multiplexed output of that module, as well as all other modules in the chassis through the SCXIbus. The analog input VIs route the multiplexed analog signals on the SCXIbus for you transparently. So, if you operate all modules in the chassis in multiplexed mode, you need to cable only one of the modules directly to the DAQ device. When an analog input module operates in parallel mode, the module sends each of its channels directly to a separate analog input channel of the DAQ device cabled to the module. You cannot multiplex parallel outputs of a module on the SCXIbus. You must cable a DAQ device directly to a module in parallel mode to access its input channels. In this configuration, the number of channels available on the DAQ device limits the total number of analog input channels. In some cases, however, you can cable more than one DAQ device to separate modules in an SCXI chassis.

When you want LabVIEW to acquire data from SCXI analog input channels, you use the analog input VIs in the same way that you acquire data from onboard channels. You also read and write to your SCXI relays and digital channels using the digital VIs in the same way that you read and write to onboard digital channels. You can write voltages to your SCXI analog output channels using the analog output VIs.

III. MEASURING TEMPERATURE WITH THERMOCOUPLES

If you want to measure the temperature of the environment, you can use the temperature sensors in the terminal blocks. If you want to measure the temperature of an object away from the SCXI chassis, you must use a transducer, such as a thermocouple. A thermocouple is a junction of two dissimilar metals that gives varying voltages based on the temperature.

However, when using thermocouples, you need to compensate for the thermocouple voltages produced at the screw terminal because the junction with the screw

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul MEO Bd. V. Pârvan Timişoara 1900 e-mail mihaela@meo.ee.utt.ro

terminals itself forms another thermocouple. You can use the resulting voltage from the temperature sensor on the terminal block for cold-junction compensation (CJC). The CJC voltage is used when linearizing voltage readings from thermocouples into temperature values.

To read the temperature sensor on the SCXI-1100, use the standard SCXI string syntax in the channels array with mtemp substituted for the channel number, as shown in the following example. To read the grounded amplifier on the SCXI-1100, use the standard SCXI string syntax in the channels array with calgnd substituted for the channel number, as shown in the following example.

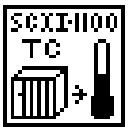


Fig.1 Connector Pane for SCXI-1100

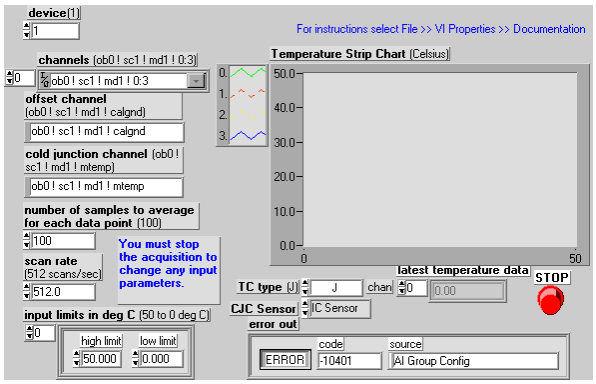

device(1)
For instructions select File >> VI Properties >> Documentation
channels (ob0 ! sc1 ! md1 ! 0:3)
offset channel (ob0 ! sc1 ! md1 ! calgnd)
cold junction channel (ob0 ! sc1 ! md1 ! mtemp)
number of samples to average for each data point (100)
scan rate (512 scans/sec)
You must stop the acquisition to change any input parameters.
input limits in deg C (50 to 0 deg C)
high limit
low limit
Temperature Strip Chart (Celsius)
latest temperature data
TC type (J)
chan
CJC Sensor
IC Sensor
error out
code
source
ERROR
-10401
AI Group Config
STOP

Fig.2 Front Panel for SCXI-1100

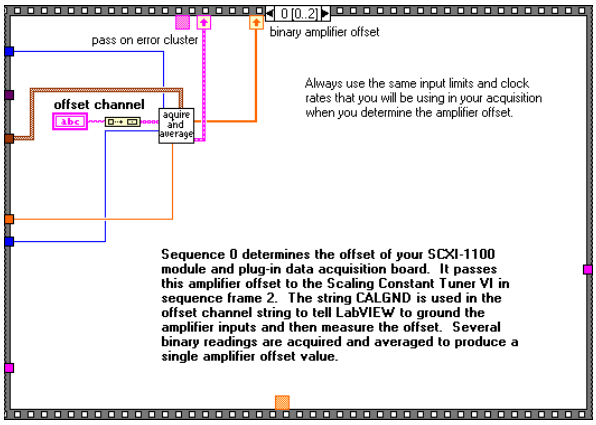

0 [0..2]
pass on error cluster
binary amplifier offset
offset channel
acquire and average
Always use the same input limits and clock rates that you will be using in your acquisition when you determine the amplifier offset.
Sequence 0 determines the offset of your SCXI-1100 module and plug-in data acquisition board. It passes this amplifier offset to the Scaling Constant Tuner VI in sequence frame 2. The string CALGND is used in the offset channel string to tell LabVIEW to ground the amplifier inputs and then measure the offset. Several binary readings are acquired and averaged to produce a single amplifier offset value.

Fig. 3.1 Block Diagram (Sequence 1) for SCXI - 1100

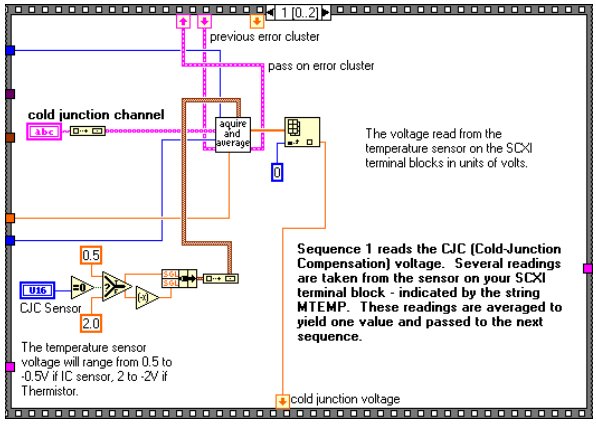

1 [0..2]
previous error cluster
pass on error cluster
cold junction channel
acquire and average
The voltage read from the temperature sensor on the SCXI terminal blocks in units of volts.
0.5
CJC Sensor
2.0
The temperature sensor voltage will range from 0.5 to -0.5V if IC sensor, 2 to -2V if Thermistor.
Sequence 1 reads the CJC (Cold-Junction Compensation) voltage. Several readings are taken from the sensor on your SCXI terminal block - indicated by the string MTEMP. These readings are averaged to yield one value and passed to the next sequence.
cold junction voltage

Fig. 3.2 Block Diagram (Sequence 2) for SCXI - 1100

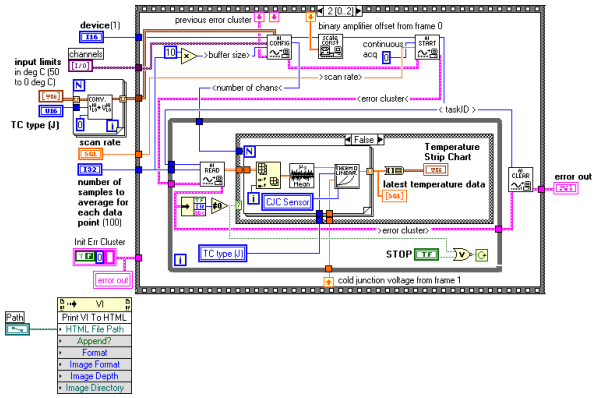


Fig. 3.3 Block Diagram (Sequence 3) for SCXI-1100

To reduce the noise on the slowly varying signals produced by thermocouples, you can average the data and then linearize it. For greater accuracy, you can measure the amplifier offset, which helps scale the data and lets you eliminate the offset error from your measurement. After measuring the amplifier offset, measure the temperature sensor for CJC. Both the amplifier offset and cold-junction measurements should be taken before any thermocouple measurements are taken. Use the Acquire and Average VI to measure temperature sensors. The main differences between the amplifier offset measurement and temperature sensor measurement are the channel string and the input limits.

If you set the temperature sensor in mtemp mode (the most common mode), you access the temperature by using mtemp. If you set the temperature sensor in dtemp mode, you read the corresponding DAQ device onboard channel. Make sure you use the temperature sensor input limits, which are different from your acquisition input limits. To read from a temperature sensor based on an IC sensor or a thermistor, set the input limit range from +2 to –2 V.

In the following we represent the Continuous Thermocouple Measurement VI which is used to read multiple temperatures from each channel. Enter your Device number, Thermocouple Type, Cold Junction

Channel, CJC Sensor, Temperature Range in Deg C, and the Channels you want to read.

Fig.4 Connector Pane for Continuous Thermocouple Measurement

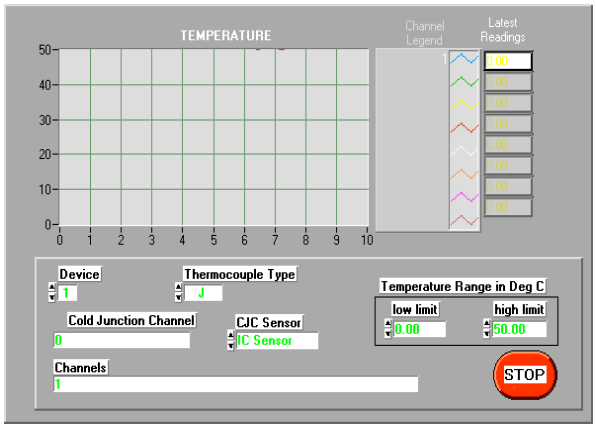


Fig.5 Front Panel for Continuous Thermocouple Measurement

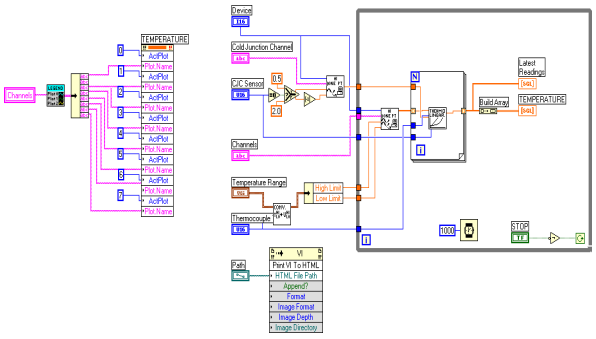


Fig.6 Block Diagram for Continuous Thermocouple Measurement

This example is build only with a data acquisition board Lab-PC+, without using a SCXI modul.
The following example presents a simulated situation of measuring the temperature in two points. Of course, this structure can be extended for measuring the temperature multipoint.

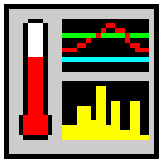


Fig.7 Connector Pane for Measuring Temperature in Two Points

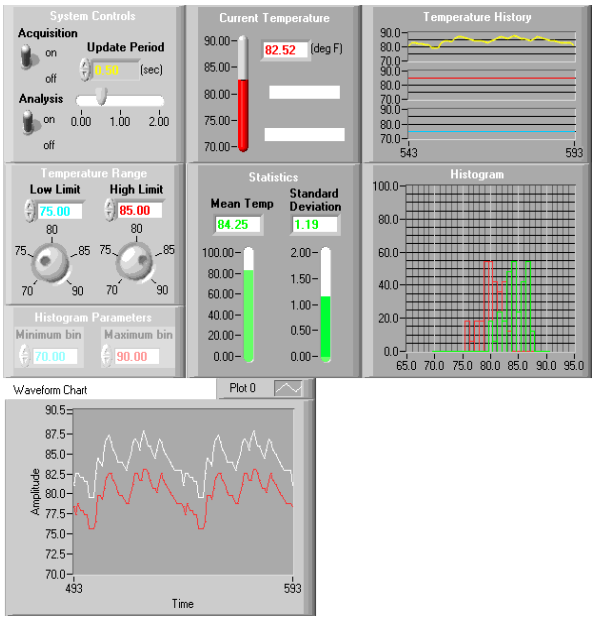


Fig.8 Front Panel for Measuring Temperature in Two Points.

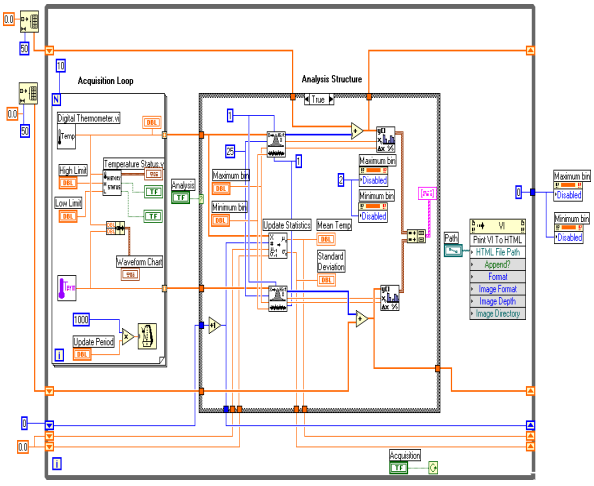


Fig.9 Block Diagram for Measuring Temperature in Two Points

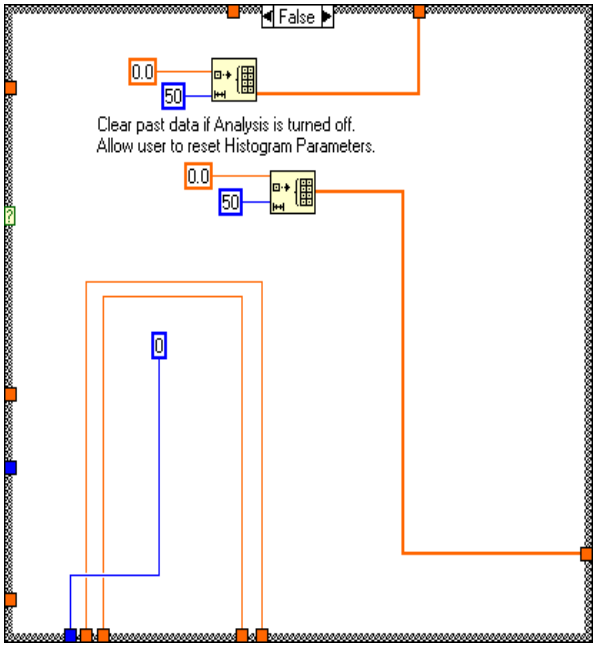


Fig.10 False State of the Case Structure

In Measuring Temperature in Two Points I didn't use Analog Input software devices, only I demonstrated with a Temperature System in Two Points a temperature aquisition and analysis application. This VI reads two simulated temperatures; alarms if one and/or the other is outside a given range; performs a statistical mean, standard deviation, and histogram of the temperature history, concerning the two points.

5.CONCLUSIONS

This paper presents three modalities of acquisition, monitoring and analysing temperature. The first example shows a situation in which a SCXI-modul, an aquisition board Lab-PC+, and Analog Input VI instruments are used. The second example is only with a Lab-PC+ board and Analog Input Virtual Instruments.
In many practical situations it is usefull to create a simulated example, that runs without failure, and only after this step it is necessary to use an aquisition board. Of course, when it is necessary to do acquisitions from transducers and sensors it is obligatory to use as well a SCXI-modul.
For all the examples I created HTML files.
File:///C/Fis.HTML/scxi 1100.html
File:///C/Fis.HTML/Trad.TermocContin.html
File:///C/Fis.HTML/Temp2pct.html,
that are very usefull in analysing and comparing virtual instruments.

7. REFERENCES

[1] L. Toma, "Sisteme de achiziţie şi prelucrare numerică a semnalelor", Editura de Vest, Timişoara, 1996.

[2] A.Ignea, "Măsurarea electrică a mărimilor neelectrice", Editura de Vest, Timişoara, 1996.

[3] National Instruments, LabVIEW User Manual, 1999.

[4] National Instruments, LabVIEW Measurements Manual, 1999.

[5] National Instruments, LabVIEW Development Guidelines, 1999.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Implementation of an AC Bridge using a Lab PC+ Data Acquisition Board

Septimiu Mischie[1], Alimpie Ignea[1]

***Abstract* –The paper presents implementation of an AC bridge based on a LAB PC+ data acquisition board. The balance of the bridge is made using the LMS algorithm. A real-time implementation of the bridge is presented and tests results obtained are incorporated in this paper.**
Index terms –balance of AC bridge, virtual instrument, adptive LMS algorithm.

I. INTRODUCTION

Usual, an alternative current (AC) bridge for impedances measurement contains four impedances. When the bridge is balanced, the unknown impedance can be computed function of the known impedances. This method has the disadvantage of manually balanced and this process can have a long time. Also, the price can be expensive because the tree impedance must be adjustable and accuracy.

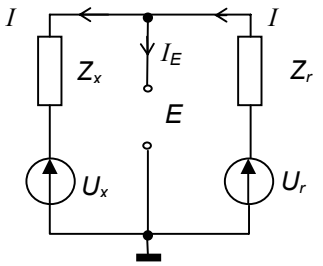


Fig.1 AC bridge with two sinewaves generators and two impedances

Fig.1 shows another version of AC bridge, namely with virtual arms [1], [2]. It contains in two arms two sinewave generators (U_x and U_r) and two impedances (Z_x and Z_r) in another two arms. If the bridge is balanced, the error voltage E=0, (I_E=0) and the next expression can be write

$$\frac{\overline{U}_x}{\overline{U}_r} = -\frac{\overline{Z}_x}{\overline{Z}_r}, \tag{1}$$

where all variables are complexe quantities. In process of balancing of the bridge, if U_r is fixed (reference voltage), U_x must be modified so E=0, and the equation (1) is satisfied. If the two sinewave generators are implemented using two digital to analog converters, and E will be measure with an analog to digital converter, the balancing process of AC bridge can be made using a computer which controls the three converters. Thus, the balancing process can be made automatically in a very short time and with a minimum of accuracy impedances.

For simplicity Z_r is chosen to be a pure resistor, Z_r=R (reference resistor). The momentane values of two sinewaves can be write as follows

$$\begin{aligned} u_r(t) &= A\sin(\omega t), \\ u_x(t) &= B\sin(\omega t + \phi) \end{aligned} \tag{2}$$

The parameters B and ϕ of the U_x generator must be modified in order to balance of the bridge.

From (1) and (2) $\overline{Z}_x$ can be write as follows

$$\overline{Z}_x = -R\frac{B\exp(j\phi)}{A\exp(j0)} = -R\frac{B}{A}\exp(j\phi), \tag{3}$$

or, if $\overline{Z}_x = R_x + jX_x$,

$$\begin{aligned} R_x &= -R\frac{B}{A}\cos\phi, \\ X_x &= -R\frac{B}{A}\sin\phi \end{aligned} \tag{4}$$

In practice there are 3 cases, function of caracter of unknown impedance; Z_x can be pure resistor, inductive impedance or capacitive impedance. Thus, if U_r, R and Z_x are fixed, B and ϕ must be modified so the next relation will be true:

$$R\overline{U}_x = -\overline{Z}_x\overline{U}_r \tag{5}$$

1. For $\overline{Z}_x = R_x$, from equations (5) and (2) results ϕ =π.

[1]S.Mischie, A.Ignea,are with the Electronics and Telecommunications Faculty, Dept. of Measurement and Optical Electronics, Bd. V. Pârvan, 1900 Timişoara, e-mail: mischie@meo.ee.utt.ro, a_ignea@meo.ee.utt.ro,

From (4) results R_x >0 and X_x = 0, so it expected.

2. For $\overline{Z}_x = R_x + j\omega L_x$, equation (5) is shown in fig.2 as phasor diagram and follows $\phi \in [\pi, 3\pi/2]$.

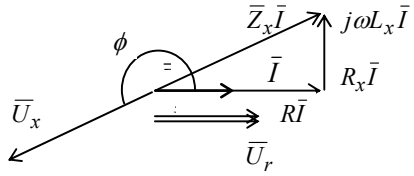


Fig.2 Phasor diagram for $\overline{Z}_x = R_x + j\omega L_x$

From (4) it follows R_x >0 and X_x > 0, so it expected.

3. For $\overline{Z}_x = R_x + 1/(j\omega C_x)$, equation (5) is shown in fig.3 as phasor diagram and results $\phi \in [\pi/2, \pi]$.

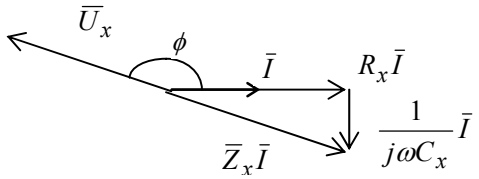


Fig.3 Phasor diagram for $\overline{Z}_x = R_x + 1/(j\omega C_x)$

From (4) it follows R_x >0 and X_x < 0, so it expected.

In this paper the AC bridge with virtual arms is implemented using a data acquisition board of National Instruments Lab PC+ type and the LMS algorithm.

II SYSTEM IMPLEMENTATION

A.Hardware Description

The block diagram of realized AC bridge is presented in fig.4. The voltages U_x and U_r are generated using two 12-bit digital to analog converters (DAC) with output range ±5V and E is acquired using a 12-bit analog to digital converter (ADC) with input range ±5V. The three converters are on the Lab PC+ Data Acquisition Board. This board is see as an input-output device for the μP (microprocessor) of the IBM-PC computer and this controlls the Lab PC+.

The Lab PC+ has 28 input-output registers which have consecutive addresses. The address of first register, or base address, was chosen 260H. The microprocessor reads or writes these registers and thus the three converters can be controlled.

B. The bridge balancing

The bridge balancing is equivalent with modifying U_x (magnitude B and phase ϕ) so E=0. For this purpose is used the LMS (Least Mean Square) algorithm. Fig.5 shows the block diagram of the adaptive filter which implements this algorithm

U_x can be write as a weight sum between a *sin* function and a *cos* function,

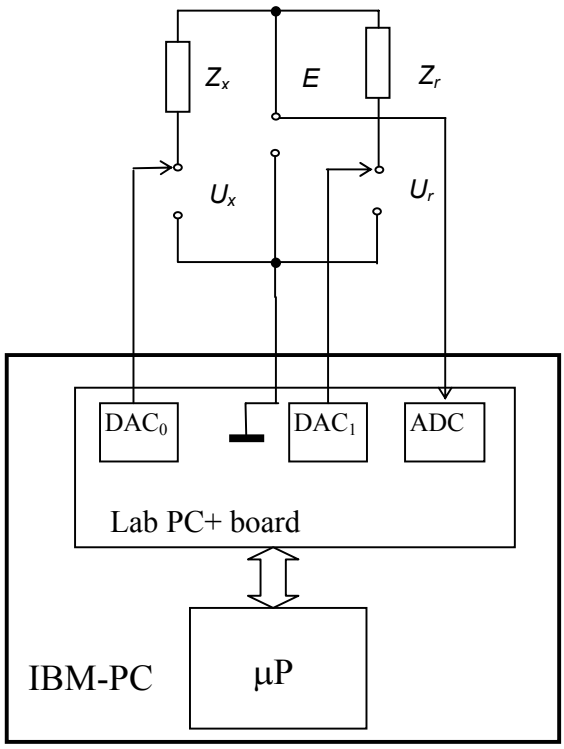


Fig.4 The block diagram of AC bridge

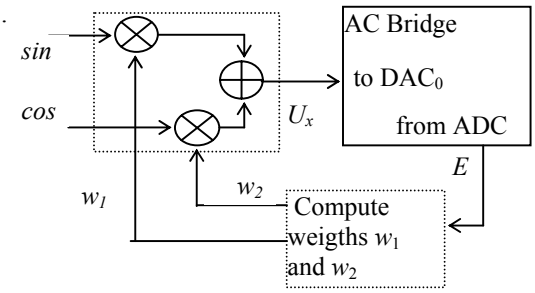


Fig.5 Block diagram of algorithm

$$u_x(t) = w_1 A\sin(\omega t) + w_2 A\cos(\omega t), \quad (6)$$

or, in complex representation,

$$\overline{U}_x = w_1\overline{U}_r + jw_2\overline{U}_r \quad (7)$$

From (5) and (7) follows that

$$\overline{Z}_x = -R(w_1 + jw_2), \quad (8)$$

or,

$$R_x = -Rw_1$$
$$X_x = -Rw_2 \quad (9)$$

The reference voltage U_r is achieved as a digital sinus

$$u_r(i) = A\sin\frac{2\pi}{n}i,\ i = 0,1,2,... \quad (10)$$

where n is the number of points per period.

The weights w_1 and w_2 at time k+1 can be calculate using the LMS algorithm as follows[4]

$$w_1(k+1) = w_1(k) + \mu EA \sin\frac{2\pi}{n}k$$
$$w_2(k+1) = w_2(k) + \mu EA \cos\frac{2\pi}{n}k \quad (11)$$

where μ is the convergence factor which controls the stability and adaptation rate.

C.Software structure

The program has been written in C language. First, the values of the U_r was calculated in a table with n locations, using (10). Next, it follows a periodic process that contains mainly the following steps (the two weights w_1 and w_2 has the initial values at 1, and the voltage of both DAC's was initially 0V):

1. Acquire one sample from E using ADC,
2. Send a 12-bit word to DAC_1 to obtain U_r,
3. Compute U_x with (6),
4. Send a 12-bit word to DAC_0 to obtain U_x,
5. Compute w_1 and w_2 with (11),

This periodic process is finished when E, is smaller than a given threshold, or the LMS algorithm has converged.

In step 3, the values for *cos* function has been obtained by reading from the *sin*-table with a n/4 offset.

The program in C-language gives values for a *sin* function in a floating point format. Then, these values hase been converted in a 12 bit two's complement format for DAC's command. Writing the least significant 8 bits and then the most significant 4 bits loads the each two DAC's.

The period of this process (T_p) is very important and must be precisely generate because:

1. The sample time T_s for acquisition of E is equal with period T_p.
2. The period T of both waveforms U_x and U_r is multiple of T_p, namely

$$T = nT_p \quad (12)$$

Thus, T_p has been achieved using a 8253 Counter-Timer which is on LabPC+ board.

III. EXPERIMENTAL RESULTS

A. Values of parameters

By programming the 8253 Counter-Timer, T_p had a value of 100 μs. Thus, follows that F_s=1/T_s=10 kHz and, because for n, number of points per period, a value of 100 is assigned, follows T=10ms or f=1/T=100Hz. That is, impedances will be measure at a frequency of 100 Hz, and for the acquisition of E, the ratio F_s/F is 100. Also, for inductive impedances measurement, because L_x had values of order of 0.1÷2H, in order to increase Z_x, frequency of U_x and U_r was 200 Hz (by appropriate programming of 8253 circuit)

Another important problems was values of impedances from the two arms of bridge and amplitude of reference voltage U_r.

First, because maximum load current of each DAC's is 2mA, follows that minimum value of sum between unknown impedance and reference impedance must be about 5kΩ.

Second, the value of reference impedance R must be choose so the bridge be near of the balance, i.e., value of E will be maximum of 5V (amplitude), in order the ADC not will be saturate. Also, due to (9), R_x and R, or X_x and R must be of approximate the same order of magnitude. If w_1 or w_2 have values very different of 1 (for exemple, highest of 2.5), that is U_x, from (6) can result highest of 5V (amplitude). From these observations, amplitude of U_r must be setting to maximum middle of output range of DAC.

Third, in order that the bridge has maximum sensibility, Z_x and R, and then, U_r and U_x must be have the same order of magnitude.

From these many constraints authors must select one middle solution. Thus: (1) value of reference resistor R is choosen function of approximate value of Z_x; (2) amplitude of U_x is choosen about to 2.5V, but depending of expected values of w_1 or w_2, can be decrease to about 1.5V.

Value of the coefficient μ from LMS algorithm must have a value in interval [0, $1/\lambda_{max}$], where λ_{max} is the largest eigenvalue of the autocorelation matrix of input sequence (the sum between a *sin* function and a *cos* function) in adaptive filter. Because values of input sequence are numbers in range (-2048, 2047), μ must have a very little value. Experimental was obtained that a value of about 10^{-6} leads to a fast convergence of LMS algorithm.

B. Obtained results

In experimental operations we have been measured resistors, capacitive impedances and inductive impedances. In table 1 are presented values which has obtained after the measure with implemented AC bridge, and for comparison, values which has obtained by measuring with another instrument. Values of amplitude of U_r, reference resistor R, and values obtained for w_1 and w_2 are presented too. It can see, that generally, errors are small, specialy for resistors, capacitors, and inductances. For serial resitors of capacitive impedances and inductive impedances the errors are higher, because theirs values are very little comparative with R, and from relation (9) w_1 is very small. Consequently, the first term in (6) will be of order of few LSB of DAC.

In fig.6 is shown variation of error voltage E. The amplitude is in LSB. It can see that after about 700 samples, or 7 periods, this voltage is approximately 0. It results that, the LMS algorithm is very fast and converged in a very short time and measuring process is a real-time.

Tab.1 Measured values

R	Measured values with AC bridge	Measured values with insruments	Weights	Magnitude of U_r [LSB]
3.38 kΩ	R_x= 6.811 kΩ	6.84 kΩ	w_1=-2.0152 w_2=-0.0003	800
3.38 kΩ	C_x=0.3064 μF R_x=110Ω	0.33 μF	w_1=-0.038 w_2=1.537	800
21.6 kΩ	R_x=7.245 kΩ	7.25 kΩ	w_1=-0.3354 w_2=-0.0001	1200
21.6 kΩ	C_x=0.3181 μF R_x=132Ω	0.33 μF	w_1=-0.0061 w_2=0.2310	1000
21.6 kΩ	C_x=0.1291 μF R_x=328Ω	0.12 μF	w_1=-0.015 w_2=0.569	1000
3.38 kΩ	L_x=0.931 H R_x=93 Ω	1.0 H 105 Ω	w_1=-0.027 w_2=-0.346	700
3.38 kΩ	L_x=1.912 H R_x=222 Ω	2.0 H 215 Ω	w_1=-0.073 w_2=-0.711	700

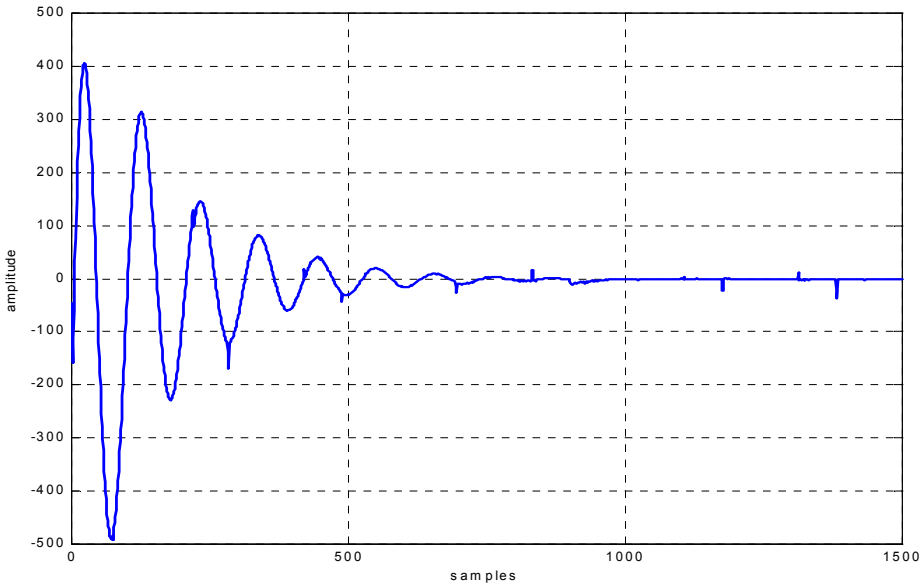


Fig.6 Variation of E voltage

IV. CONCLUSIONS

This paper presented implementation of an virtual AC bridge for impedance measurement using an IBM-PC which contains a data acquisition board with two DAC's and one ADC. This virtual instrument works in real-time using a C-language program. Experimental tests were carried out on resistors, capacitive impedances and inductive impedances. The obtained results have very small errors.

REFERENCES

[1] M. Dutta, A.Rakshid, S.N.Battacharyya and J.K.Choudhury, "An application of the LMS adaptive algorithm for a digital AC bridge," *IEEE Trans.Instrum.Meas.,*vol.36, pp.894-897, 1987

[2] Selim S. Awad, Natarajan Narasimhamurthu and Wiliam H. Ward, "Analysis, Design, and Implementation of an AC Bridge for Impedance Measurements, *IEEE Trans.Instrum.Meas.,*vol.43, pp.894-899, 1994

[3]L. Angrisani, A.Baccigalupi and A.Pietrosanto, "A digital Signal-Processing Instrument for Impedance Measurement" *IEEE Trans.Instrum.Meas.,*vol.45, pp.930-934, 1996

[4]B. Widrow, S.Stearns, "Adaptive Signal Processing", Prentice Hall, New Jersey, 1985

[5] Lab-PC+ User Manual, National Instruments, 1994

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Considerations Concerning the Practical Behaviour of a Digitising Oscilloscope in Different Signal Recording Modes

Daniel Belega [1] and Septimiu Mischie [1]

Abstract – In this paper is analysed the practical behaviour of the digitising oscilloscope (DSO) – HM407 in different signal recording modes: Refresh (RFR), Average (AVM) and Envelope (ENV). In RFR and AVM modes were experimentally evaluated the dynamic performance of the HM407 by the method proposed [1]. It is presented some important conclusions concerning the dynamic performance of the HM407 obtained in these modes. For different input signal types were shown the signals acquired in ENV mode. The signal acquired in this mode permits a qualitatively appreciation of the input signal amplitude and frequency variations.
Index terms: Testing of waveform digitizer, software for IEEE standards 1057 and 1241.

I. INTRODUCTION

The acquisition process of an analog signal is often made by means of a digitising oscilloscope (DSO). The actual DSOs have, in addition, some signal recording modes, like for example: computation and displaying the average value of several signal recordings, displaying the envelope curve of the acquired signal, peak detector etc. [2], [3]. For efficiently using of such DSO it is very useful to know, based on the experimental results, the behaviour of DSO in each signal recording mode. Also, it is very important to know with high precision the dynamic performance of the DSO obtained in different signal recording modes.
In this paper is experimentally analysed the behaviour of an actual DSO – HM407 in different signal recording modes: Refresh (RFR), Average (AVM) and Envelope (ENV). HM407 is an 8-bit DSO with 40 MHz analog bandwidth, a 100 MS/s maximum sampling rate, and a maximum record length of 2048 samples [2]. Also, HM407 has only an interface - the serial interface RS232C - for the communication with a PC.
One of the most important dynamic parameter of a digitizer is the signal to noise plus distortion ratio (SINAD). Using the method proposed in [1] was estimated the SINAD of HM407 in the RFR and AVM modes. This method leads to very accurate SINAD estimates when the number of samples is obtained from a number of input signal cycles higher than 20 [4]. This method is presented in Appendix A.
For different input signal types the signals acquired in ENV mode are qualitatively appreciated.

II. THE EXPERIMENTAL SETUP

The experimental setup used for analysing the practical behaviour of the HM407 in RFR, AVM and ENV modes is shown in Fig. 1.

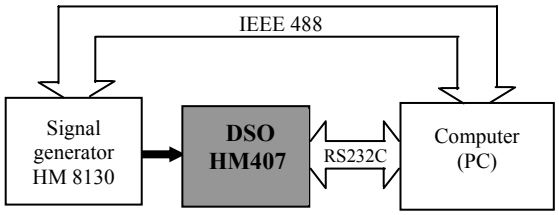


Fig. 1. Experimental setup.

The IEEE Standard 1057 suggests the use of high purity sinewave signals for testing the digitizers [6]. The sinewave generator employed for this work was the programmable function generator HM8130 [7]. The sinewaves used for testing the HM407 in RFR and AVM modes were characterised by frequencies situated in the range 1÷20 kHz and peak-to-peak amplitudes smaller than 6 V. The dynamic performance of these sinewaves, obtained by testing, permits an accurate dynamic evaluation of a waveform digitizer which have an effective number of bits (ENOB) smaller or equal with 8 bits.
In the analysis of the practical behaviour of HM407 in ENV mode were used different input signal types.
The PC via an IEEE 488 interface established the parameters of the signals used.
At the DSO was selected the AC input coupling.

[1] Faculty of Electronics and Telecommunications,
Dept. of Measurements and Optical Electronics,
e-mail: belega@meo.ee.utt.ro

Data records acquired by the DSO were transferred to the PC via the RS232C interface.
The PC contains the program which implements the acquisition process and the program package presented in [4]. In this program package is implemented the method proposed in [1]. For this method the program package provides a number of graphical pages.
The program that implements the acquisition process is written in C language.

III. DYNAMIC TESTING OF HM407 IN RFR AND AVM MODES

To ensure that the HM407 in RFR and AVM modes is correctly tested the sinewave generator was adjusted so that the peak-to-peak amplitude of the sinewave detected by the HM407 coincides or is slightly smaller than its amplitude range.
Different settings of the HM407 time base and vertical gain were tested. Several acquisitions were made at different test frequencies. In all the situations analysed, in both modes, the number of samples used was M = 2048, and was obtained from a number of input sinewave cycles higher than 20. Also, in all the situations analysed, in both modes, was used the 4-term maximum decay window [8]. Fig. 2 reports the SINAD estimates obtained in RFR mode at a given frequency. Each plot refers to a different HM407 vertical gain.

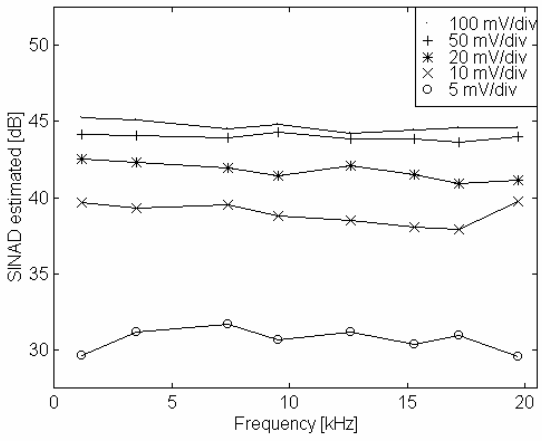


Fig. 2. SINAD estimates as a function of frequency for different HM407 vertical gains.

From Fig. 2 results that the SINAD estimates remains relatively constant in the considered frequency range. For an ideal 8-bit digitizer SINAD = 49.92 dB. Thus, from Fig. 2 results that the electrical noise within the digitizer is higher than the quantization noise, especially for small vertical gains.
Fig. 3 shows the dynamic performance obtained in RFR mode for a vertical gain of 50 mV/div and with a HM407 sampling rate f_s = 400 kHz. The sinewave test signal was characterised by a frequency f_{in} = 9.5 kHz and an offset V_0= 0 V. Fig. 3 was carried out using the program package presented in [4].

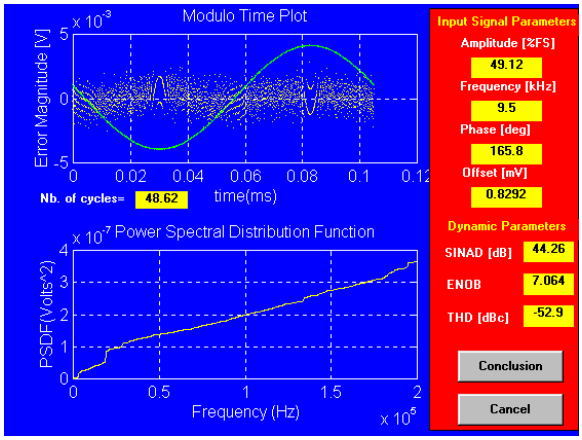


Fig. 3. The dynamic performance obtained in the situation fs = 400 kHz and f_{in} = 9.5 kHz.

In top graph of Fig. 3 are presented the "Modulo Time Plots" [9] of the output signal and of the residual error [4]. The amplitude of the output signal cycle plotted has no meaning. This cycle is displayed to show the phase of each residual with respect to the output signal. By means of these plots is characterised the acquisition process [9]. In bottom graph of Fig. 3 is represented the power spectral distribution function (PSDF) of the residual error. This plot put very clear in evidence the distortions, which affects the output signal [10]. In this graphical page are presented the dynamic parameters of the digitizer: SINAD, ENOB and total harmonic distortions (THD). ENOB was estimated by the method proposed in [1] (eq. (A.4) from Appendix A). THD was estimated by the Interpolated FFT algorithm [5]. Also, are presented the parameters of the best sine fit that corresponds to the digitizer output signal.
When M is a power of 2, for correct estimation of the dynamic performance it is necessary to overcome the situations in which the number of input sinewave cycles is equal or very close to a number which is an integral power of 2. In these situations are not tested all the digitizer levels. The performance shown in Fig. 4 were obtained in the same conditions like the ones used in the case of Fig. 3 but with f_s = 200 kHz and f_{in} = 6.25 kHz.

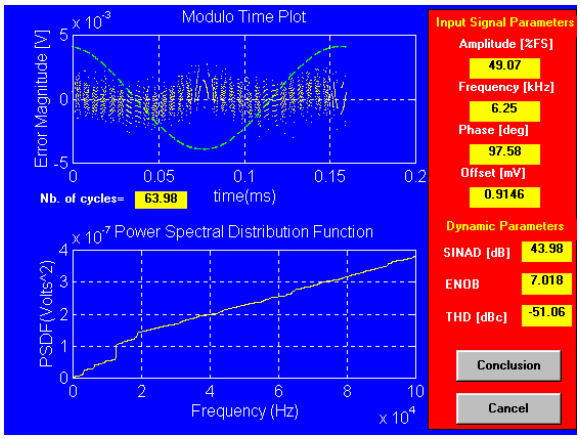


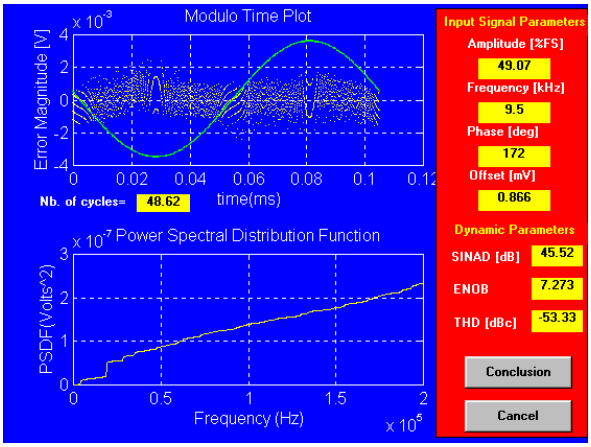
Fig. 4. The dynamic performance obtained in the situation in which the number of input sinewave cycles is 64.

From the "Modulo Time Plots" is clearly evident that in this situation remains digitizer levels which are not tested. If are not tested all the digitizer levels we are not sure that the performance obtained rest the same. Therefore, these performance corresponds only to the digitizer levels tested and not to the digitizer.

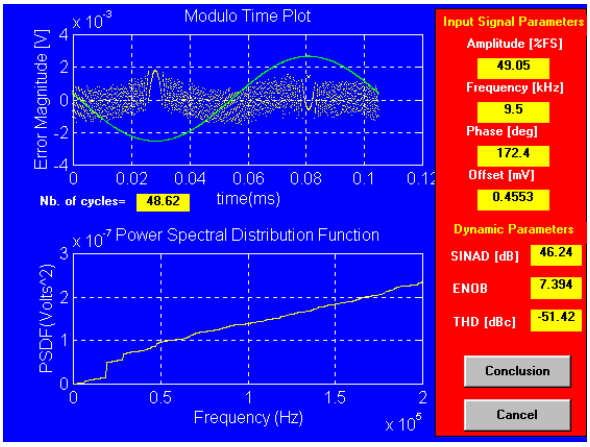
Fig. 5 shows the dynamic performance obtained in AVM mode for a vertical gain of 50 mV/div and with a HM407 sampling rate f_s = 400 kHz. The sinewave test signal parameters were f_{in} = 9.5 kHz and V_0 = 0 V. In Fig. 5(a) the number of recordings used for averaging was M_r=128 and in Fig. 5(b) was M_r=512.

By comparison of the "Modulo Time Plots" of the residual errors presented in Fig. 5 with the "Modulo Time Plot" of the residual error shown in Fig. 4 results that the averaging reduces the noise. As consequence, in AVM mode, SINAD and ENOB increases and so the performance of the HM407 increases.

From Fig. 5 results that the dynamic performance increases when M_r increases. Therefore, we recommend the use the maximum available value for M_r.



(a)



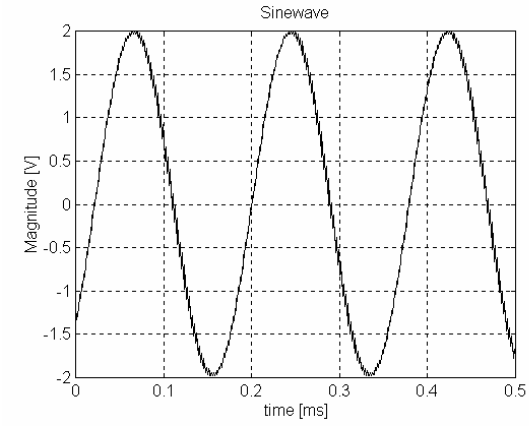
(b)

Fig. 5. The dynamic performance obtained with: (a) M_r=128; (b) M_r=512.

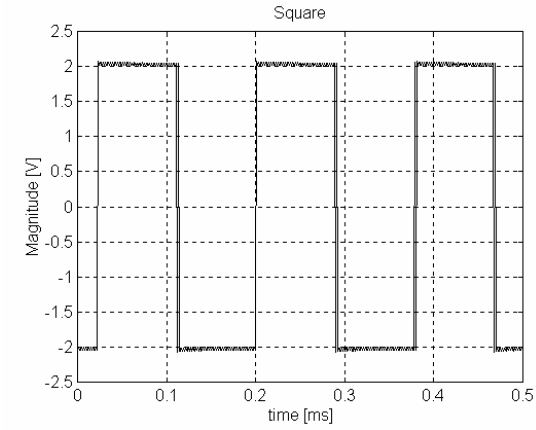
IV. THE BEHAVIOUR OF HM407 IN ENV MODE

Fig. 6 shows the signals acquired in ENV mode for different input signal types as a function of time for a vertical gain of 500 mV/div and with a HM407 sampling rate f_s = 1 MHz. The input signals were characterised by an amplitude A = 4 V, f_{in} = 5.6 kHz and V_0 = 0. The number of samples acquired was M = 2048. Both plots presented in Fig. 6 were obtained by means of MATLAB program.

In both plots shown in Fig. 6 is evident that the input signal amplitude and frequency vary. These plots permits only a qualitatively appreciation of the input signal amplitude and frequency variations.



(a)



(b)

Fig. 6. The signals acquired as a function of time in the case of a: (a) sinewave input signal; (b) square input signal.

V. CONCLUSION

This paper analyses the practical behaviour of an actual DSO – HM407 in some signal recording modes: RFR, AVM and ENV.

Based on the results of the dynamic testing of HM407 in RFR and AVM modes were reached some important conclusions:

- in both modes, when the number of samples acquired is a power of 2, for testing all the digitizer levels is necessary to overcome the situations in which the number of the input sinewave cycles is

equal or very close to a number which is an integral power of 2;

• in AVM mode the best performance are obtained in the case of using the maximum available value for M_r.

The ENV mode permits a qualitatively appreciation of the input signal amplitude and frequency variations.

APPENDIX A
DESCRIPTION OF THE METHOD PROPOSED IN [1]

Assume a pure sinewave acquired by the digitizer under test at a sampling frequency f_s; the number of samples acquired is M. The data record obtained is $y(m)$, m=0,1,…, M-1.

For estimating the dynamic performance of the digitizer the method proposed in [1] supposes the following steps:

step 1: Is determined the best sine fit corresponding to the digitizer output data signal by means of interpolated fast Fourier transform (Interpolated FFT) algorithm [5]

$$\hat{y}(m) = \hat{A}\sin\left(2\pi m\frac{\hat{f}_{in}}{f_s} + \hat{\varphi}\right) + \hat{d}, \quad m = 0,1,\ldots,M-1 \tag{A.1}$$

where $\hat{A}, \hat{f}_{in}, \hat{\varphi}, \hat{d}$ are the amplitude, frequency, phase and offset of the best sine fit, estimated by Interpolated FFT algorithm.

step 2: Is computed the residual error $err(m)$ between the digitizer output data signal $y(m)$ and the best sine fit $\hat{y}(m)$

$$err(m) = y(m) - \hat{y}(m), \quad m = 0,1,\ldots,M-1 \tag{A.2}$$

step 3: SINAD and ENOB are estimated by the following relationships:

$$SINAD_{est}[dB] = 20\log_{10}\left(\frac{\hat{A}/\sqrt{2}}{rms(err)}\right) \tag{A.3}$$

$$ENOB_{est} = n - \log_2\left(\frac{rms(err)}{ideal\ rms(err)}\right) \tag{A.4}$$

where

n is the resolution of the analog-to-digital converter (ADC) of the digitizer,

$ideal\ rms(err) = \frac{q}{\sqrt{12}}$, q is the ideal code bin width of the ADC.

REFERENCES

[1] D. Belega, "Accurate dynamic characterization of digitizing signal analyzers," Proceedings of the Sympozium on Electronics and Telecommunications (ETC 2000), vol. II, pp.77-80,Timişoara, 2000.

[2] Hameg Instruments, Manual for Oscilloscope HM407, 1996.

[3] Tektronics, User Manual for TDS210 Digital Real-Time Oscilloscope, 1998.

[4] D. Belega, "Contributions at the testing of analog-to-digital converters," Ph. D. thesis, Universitatea "Politehnica" Timişoara, 2001.

[5] C. Offelli, D. Petri, "Interpolation Techniques for Real-Time Multifrequency Waveforms Analysis," *IEEE Trans. Instrum. Meas.*, vol. 39, pp. 106-111, Feb. 1990.

[6] S. J. Tilden, T. E. Linnenbrink, P. J. Green, "Overview of IEEE-STD-1241. Standard for Terminology and Test Methods for Analog-to-Digital Converters," Instrum. and Meas. Technology Conference, Venice, vol. 3, pp. 1498-1503, May 1999.

[7] Hameg Instruments, Manual for Function Generator HM8130, 1996.

[8] A. H. Nutall, "Some windows with very good sidelobe behaviour," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 84-91, Feb. 1981.

[9] F. H. Irons, D. M. Hummels, "The Modulo Time Plot - A Useful Data Acquisition Diagnostic Tool," *IEEE Trans. Instrum. Meas.*, vol. 45, pp. 734-738, June 1996.

[10] J. J. Blair, "A Method for Characterizing Waveform Recorder Errors Using the Power Spectral Distribution," *IEEE Trans. Instrum. Meas.*, vol. 41, pp. 604-610, October 1992.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

New Interactive Multimedia Application Model Using Objects Storage in a Database

Muguraş D. Mocofan[1], Corneliu I. Toma[2]

Rezumat – Lucrarea prezintă un model de aplicaţii multimedia ce permite utilizarea celor mai moderne tehnici de programare: programarea orientată pe obiecte şi cea orientată pe aspecte. Modelul propune un mod de structurare al aplicaţiilor multimedia ce permite o bună reutilizare a codului scris şi în alte aplicaţii, şi stocarea obiectelor utilizate în baze de date.
Cuvinte cheie: multimedia, tehnici de programare, aplicaţii interactive.

I. INTRODUCTION

This paper present a new interactive multimedia application model witch combine the advantages of object oriented programming with aspect programming. The model proposes a structure for multimedia applications with a good reutilization of code in other applications and objects storage in databases.

II. THE ARHITECTURE OF MULTIMEDIA APPLICATION MODEL

In the multimedia applications model architecture are four modules having a very strong interaction. Every module is composing by other small modules, and these small modules can be used very easy in other applications.

I use four categories of modules for the application kernel. The application kernel, generally is a specific part for a multimedia application, is not possible to reutilize the code, and for users is a close part (don't accept modifications). This kernel is "the manager" of entire application. He use the functions, procedure, routines witch are stored in modules. In many cases, the module is the equivalent of a functions library. One of the kernel jobs is to verify and to validate all the new functions implemented by the users. The architectures modules categories are:

- Objects processing functions;
- User interactions;
- Input/Output Operations;
- Integrated technologies.

The multimedia applications model architecture is presented in Fig. 1.

A. Objects processing functions

For *Objects processing functions* are two main categories:

- *Temporal processing* – the temporal aspect of objects is used in processing.
- *Objects processing* – for this category are two types of objects processing:
  - Single object processing;
  - Multiple objects processing.

In a multimedia application, we run in the same time different sounds, video sequences and animations. The applications run correctly when are free hardware resources. Is necessary to manage very well the application and the hardware resources, this can be realized by many temporal references. The *Time_Line* routines must to by very simple and easy to use (for a user with medium knowledge in the field of programming). The main jobs of the *Time_Line* routines are:

- Management for time constraint objects and for the hardware resources used for playing that.
- Verification if the hardware resources are free to use. If is not free for use, is necessary to stop playing the current object.
- Temporal references list construction for effects and transitions between objects.
- Management of temporal references jumping in collaboration with sub-modules *GUI* and *Events*.

Here, is useful to use the aspect programming techniques [1]. The very good results are achieved if are mixed the object oriented programming techniques for multimedia objects description and aspect programming techniques for temporal management of the events.

Object Processing Functions – for reducing the space that a multimedia application requires in the context of increasing the power of calculation, I propose the utilization of "on-line processing" concept. That

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii, Bd. V. Pârvan, Timişoara, 1900, e-mail: toma@hermes.ee.utt.ro
[2] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii, Bd. V. Pârvan, Timişoara, 1900, e-mail: mugur@mail.opendrum.utt.ro

suppose the processing of the objects is made during the running of application.

So, it can be used only one physical object in many apostasies (in several scenes it can wear a variety of forms - it can be scaled, it can be rotate or deformed, etc.) during the running of an application.

If the application uses a very large number of objects witch suffers the same processing methods, handling this objects can be simplified by using the idea of processing directly the information and not store it.

We can meet two situations:

- the new generated object can be stored (the handler can personalize the multimedia objects starting from a model object);
- or the object can be destroyed after utilization (if we need it again the object can be regenerated).

If the processes are very complex, there is the risk of loosing a lot of time with calculation, and the result cannot be obtained instantaneous. The processing functions have a big diversity and depend of the element that is processed.

Sometimes its implementation can be very heavy. I recommend structuring of the elements, in a library that contains processing functions, which can be accessed later.

For users, can be allowed the personalization of these functions, this is realizable by modification of the processing parameters. The functions in this library use generally only one object during the processing part.

Transitions – during one multimedia application, you don't know the moment in witch the user decides to go to another scene of the application.

To realize a very interesting switch between scenes, frequently are used transitions or effects between objects.

When in those scenes are objects like videos and animations, those contain different positions in each moment, that's why you cannot recalculate the transition between the two scenes (the transition between the objects from the two scenes).

The processing part it will be done in the moment decided by the user, when he wants the transition.

In this case it will be a process *on-line,* witch will use the system resources where the application is running, and we will not talk about storing the results of the process.

For a new utilization of the processes (effects and transitions) I recommend the organization of a library with distinctive processing functions *Transitions.*

Is necessary for these functions, to exist a transparency of processing parameters, in this way the users succeed to personalize their applications.

Also is good that for experimented developers of multimedia applications to exist instructions on the way to implement new functions, witch later can be distributed in new ways of multimedia applications area.

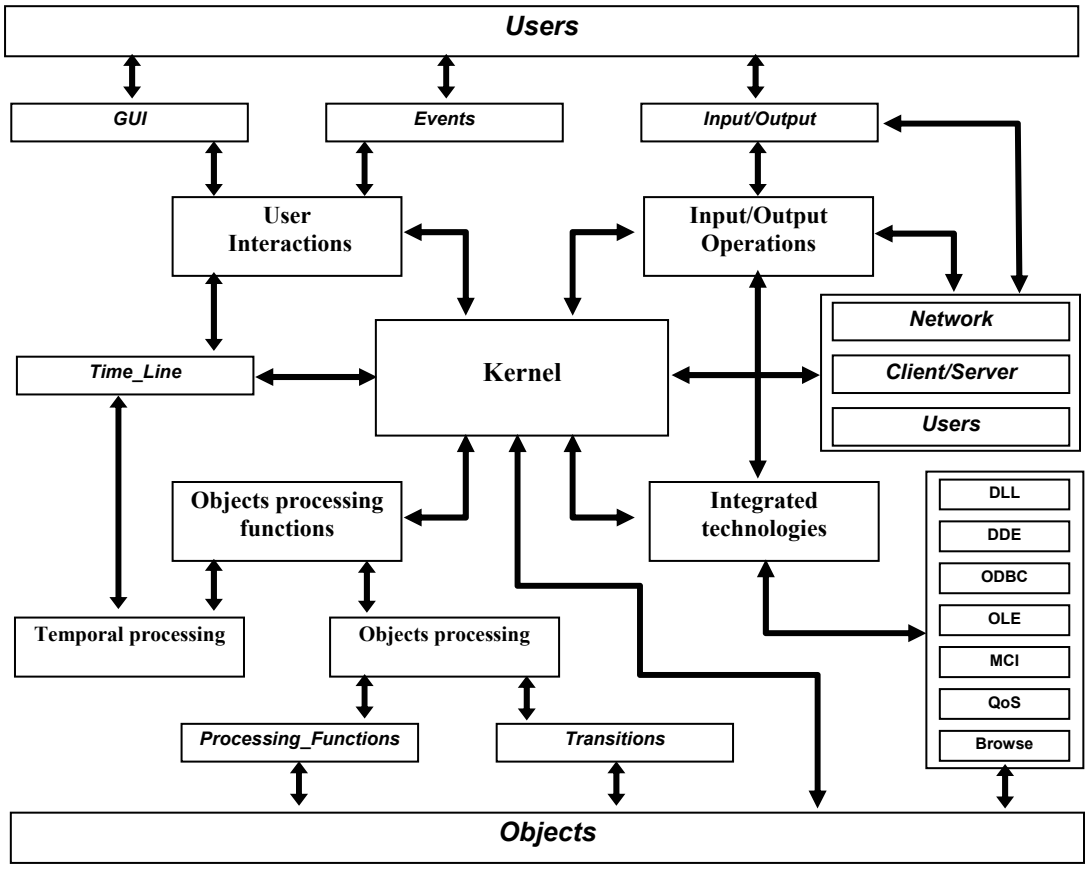


Fig. 1 Multimedia application model architecture

B. User interaction

This module is characterized through a graphical interfaces and events generated from interactions between application and user.

GUI – the graphical interface with the user is made from different graphical elements, objects witch offers interactivity between user and application (buttons, links, images, little animations, etc.). These objects have attached routine witch start when one defined event appears in a library of functions, methods and events *Events.*

Events – The events system modify the way of application run, depends on the actions made by the user in the application. This makes an event, and for them we will have a specified action. It is very important that we define correctly the system of events witch might appear and also the actions witches are realized after this.

It is necessary that we define one hierarchy of priorities in execution:

- in case of mixing over the events in the same time;
- in the case in witch one event might provoke more than one action;

C. Input/Output Operations

Input-Output Operations contains also one component witch permits the network work, the development of services client/server and a system of user validation.

Input-Output Operations represent the way of interaction between the application and the out world (especially with the hardware resources of the computer system on witch the application runs)

We can found these functions in all the programming languages. I propose the writing of small routines witch can verify the peripheral state before their utilization, the release of some occupied resources because of some damage and the determination of peripheral performances.

In the relations with the functions from *QoS* library – checking services for the quality of the provided service – (the check for a good run of a multimedia application on a certain system) it's imperious the introduction of a routine for error detection that appears in the functionality.

Together with *QoS functions* there will be generated messages witch will inform the user on the eventual problems that might appear and proposes the remove of problems.

Network – because now is a big variety of network types, and their development is continuous, we will check for the beginning the type of existing network, after that we will appeal to the specific routines for that types of network.

This kind of structure would permit an easier adaptation of the application depending on the hardware and software resources of the network.

My recommendation, for the *Network* routines, is to have possibilities to verify the network level of traffic. Is necessary to inform the user about this level of traffic and to propose different solution (if are some problems).

Client/Server – many applications use for communication implementation of client/server services. These kinds of technologies are useful for job distribution in a heterogenic network, to enhance the speed of processing [2].

Users Management – if an application is accessed by many users in the same time, or sequential the routines for this module most to offer a management of users.

D. Integrated technologies

This module contains libraries dedicated to working with databases (ODBC), integration libraries of the objects realized with other applications (OLE technologies), implementation of different kinds of services client-server libraries (DDE), with optimization functions for different equipments inside PC (MCI functions for sound, video and CD-ROM); evaluation libraries of quality service offered by the application on hardware resources by the user; function libraries witch allowed searching object and resources witch are used in the multimedia application; etc.

Dynamic Data Exchange (DDE) allowed the construction of client-server applications. DDE allows data actualization in other application and makes easier the utilization of remote commands.

Dynamic Link Library (DLL) use the extern files that contains routines and methods witch are used by Windows applications.

Object Linking and Embedding (OLE) is a techniques witch allows creation of objects and this objects can be exported and reused in other multimedia applications.

Open Database Connectivity (ODBC) - Microsoft proposed the concept for generalizing the way of communications with databases.

I suggest a generalization in using this concept, so, the multimedia applications can use this way of connecting for accessing their objects from a databases.

The ODBC technology allows a commune interface for accessing heterogenic SQL databases. SQL language is a standard of accessing databases. This interface gives a higher level of interoperability; one application can access a variety database system management witch use SQL language. This allows users to realize and distribute a client-server application, without restrictions from a system databases management [3].

MCI - an operating system has to allow the use of multimedia resources for various applications witch runs in the same time. Is necessary that their data stream to be synchronized, and for that you must use MCI functions.

QoS – Quality services – this module describes the needs of a multimedia application from the point of view of components functionality [4].

I think that the model of a multimedia application has to contain evaluation elements of the hardware resource performances.

QoS module has to realize an initial testing of the equipments that is going to run. Is necessary to inform the user about how the application will run in these conditions (eventually it may propose solutions).

The biggest problems in multimedia applications are the synchronization constraints of different types of media witch runs simultaneous.

I propose that the next parameters witch describe the quality of a service (of the way the program runs in a hardware environment) to be verified and respected by the programmer [5]:

- The rate of application run speed;
- The level of hardware utilization;
- The level of jitter;
- The temporal cumulate errors;
- The rate of environment errors;

Browse functions – because multimedia applications are becoming more and more complexes (using more than 100.000 objects) is necessary to introduce a new module witch contains browsing functions.

It is very difficult to go thru the entire application to find information, that's why, we prefer automatically solutions for searching. For this kind of applications is necessary to have an objects search functions library. This routines are very complexes, sometimes are necessary to implement also content-based methods.

A correct structuring of the objects and also the implementation of the searching functions, offers many advantages to the users, increasing the attractiveness of the application.

The *Browse* library has to be open for new developments.

III. MULTIMEDIA OBJECT DESCRIPTIVE MODEL

Each multimedia object has a value, a logical structure and an interpretation of the content for that value. The value of a multimedia object, his logical structure and the model of representation and storages must be described in the multimedia object's model.

The multimedia-describing model contains information that is referred to the logical composition of this object, the ways of synchronization and temporization for the components and the necessary parameters for displaying.

The model of interpretation has to describe exactly the real world thru the sub-objects that are composing it and thru the relationship between them and between each sub-object and the real world.

One multimedia object may be composed with other multimedia objects or with simple elements (un-composed multimedia objects).

One multimedia object may be composing with the elements:

- Texts
- Images
- Sounds
- Videos
- Graphics
- Animations
- Scripts

The model of description can be synthesized like this:

- A number of simple or complexes sub-objects compose each multimedia object, witch is organized sequentially, parallel or in a mix mode.
- The elements from the object structure can take a value, witch can define an instance of the object. It is unnecessary for the object to have defined values for all of its components. We use the heritage concept from object-oriented programming.
- For a more real description of the real world it can be established a semantic interpretation for each component of a multimedia object.
- The semantic description became usefully in the case of searching different objects in a multimedia application or when the number of objects is very large. Is necessary to have a structure of the elements based on semantic concepts.

Next, I will present the parameters that are considered very important to me for describing a multimedia object and simple elements. I have organized object descriptions in a structure of classes that has close properties to the classes that we meet in the case of object-oriented programming.

A model of multimedia description is presented in Fig. 2

For an efficient description of multimedia objects I proposed a describing parameters structure:

1. General parameters – that are describing the object, like object ID, storage physical location of the object, a flag witch indicate if the object is simple or complex and information's about element structure of the object.
2. Semantic parameters – witch describe the object thru key words.
3. Type objects parameters – these parameters describe the object, its dimensions, the way of compression, the application that generated the object (useful for example in: the number of colors, dimensions, resolution, images, etc.).
4. Global parameters – witch describe the content of objects thru many features. These parameters are useful in content database searching.
5. User parameters – specified by each user and useful only in some applications.

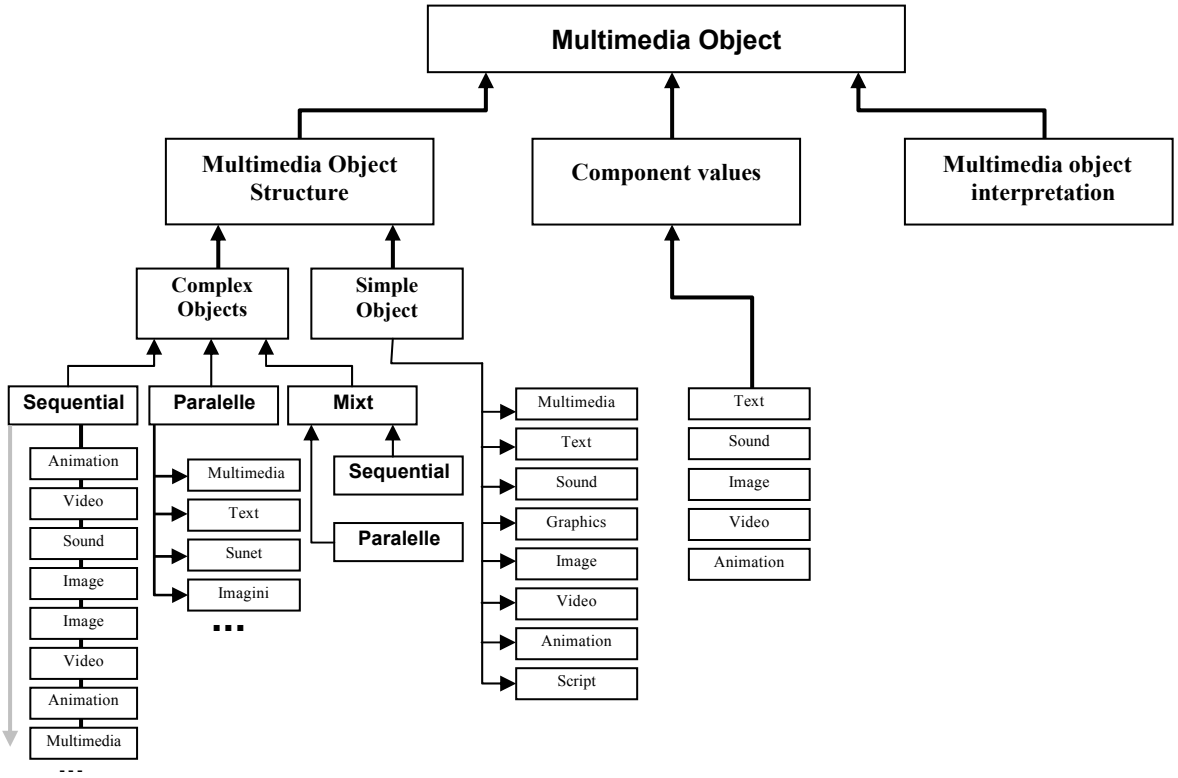


Fig. 2. Multimedia Object Descriptive Model

After describing the object follows the description of the primary elements witch are compose the object. The general parameters of description of the primary elements are:

Primary_Element

Parameters:

- **Primary_Element Identification**
- **Primary_Element General Information's**
- **General Attributes**

Primary_Element_Global_Features

Parameters:

- **Primary_Element Description**
- **Specify Information's**
- **Attributes for Content Description of Object**

Primary_Element _User_Features

Parameters:

- **User ID**
- **Primary_Element Details**
- **User Personal Information**
- **User Attributes for Content Description of Object.**

Fig. 3 Primary Element Description

IV. THE MULTIMEDIA APPLICATION MODEL CARACTERISTICS

The general characteristics of this model consist in:

- the possibility to reutilization of the elements developed previously;
- the possibility for any user to introduce new functions;
- possibility to development a old application;
- higher level of portability.

In concordance with the actual needs, my model present:

- Optimizations from the structure point of view, witch is open for an ulterior development. I use for the primary level of structure modules. For all modules is possible to define new sub-modules. A sub-module accept functions, methods and routines, is possible to grow the number of that in the future (The user can add new functions, methods and routines).
- Is possible to reuse some parts of an old application.
- Object oriented implementation for structure of used objects in applications.
- Aspect oriented implementation for the temporal aspect of multimedia objects and applications.
- If the developers of multimedia environment use the proposal model, is possible for the users to develop the functions libraries and to transfer these between applications.
- The model propose a strong system for the quality of services. A special module QoS is developed for this.

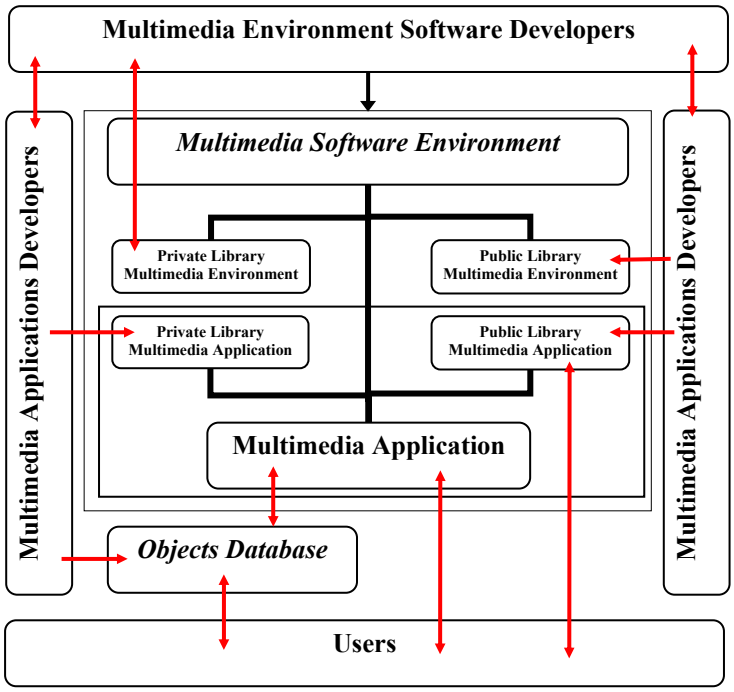


Fig. 4 Multimedia Environment in the model context

- For the objects storage, can be utilize a standard database using a generalization of the ODBC concept.
- This model realizes a dynamic content for multimedia applications. The pages included in applications are generated dynamic from a database (in real time), this pages exist if the application run, and are generated in concordance with the user options.
- Using the client/server technology, many users' acces the application in the same time using remote commands.

I propose to include in every multimedia environment tools for creation of new functions library. In this way is possible to develop the multimedia software environment.

V. CONCLUSIONS

In this days is very important to have an open environment for multimedia applications development, because are many application developers with a good experience in this field, and this experience can be use for the functions library development (Fig. 4).

From design point of view, the time necessary to realize a good concept and the modeling for application is much longer. From implementation point of view, the time for creation is in this case, much shorter. The global time spend to realize an entire multimedia application is generally shorter. If is used the real time content generation concept for pages, the time of physical implementations degrease very well.

For example, to realize a multimedia presentation for a company witch produce hundred goods is necessary to make hundred pages, one page for every product. If all the information's about the product is stored in a database, the implementation task for all pages is to make the model for a single page, and after that to connect this page to database, is not necessary to realize hundred different pages. In this dynamic way the reduction of the implementation time is considerable. In plus, the size of the executing file is smaller, because the object are loaded only when are used.

This paper present a new interactive multimedia application model witch combines the advantages of objects oriented programming with aspect programming. The model proposes a structure for multimedia applications with a good reutilization of code in other applications and objects storage in databases. Using this model is possible to develop very performing multimedia applications.

BIBLIOGRAFIE

[1] Kiczales G., Hilsdale E., Hugunin J., Kersten M., Palm J., Griswold G. W. – "Getting Started with AspectJ", Raport of Defence Advanced Research Projects Agency F30602-C-0246, 2001

[2] Mocofan M., Pescaru D. – "Parallelisation of a textured colour image segmentation alghorithm", Transactions on Automatic Control and Computer Science, Fourth International Conference on Technical Informatics - "CONTI 2000", 12-13 October, Timişoara, Romania, pp 139 – 144, 2000.

[3] Khoshafian S., Baker B. – "Multimedia and Imaging Databases," Ed. Morgan Kaufmann Publisher, San Francisco, California, 1996.

[4] Ghinea G., Thomas J. P. – "QoS Impact on User Perception and Understanding of Multimedia Clips", Proceeding of The 6-th ACM International Multimedia Conference, Bristol, England, 12-16 September, pp. 49-54, 1998.

[5] Mocofan M., Pescaru D. – "Aplicații interactive multimedia. Programarea OpenScript", Ed. Politehnica, Timişoara, 2001.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Optimised Homotropic Structuring Element for Handwriting Characters Skeleton

Ioan Şnep[1], Dan L. Lacrămă[2], Corneliu I. Toma[3]

Abstract – This paper is focused on the study of skeleton transformations involved in the hand printed & handwritten documents' images preprocessing. A new homotropic transformation structural element is proposed in order to optimize the sequential thinning procedure: reduced number of thinning cycles and less coarse skeleton lines.

Keywords: Hand-written Character Recognition, Java Programming.

I. INTRODUCTION

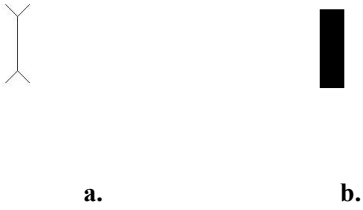
The skeleton procedure (i.e. the reduction of the analyzed characters to one pixel thick lines and curves) is one of the most important steps in the typed & hand written images' preprocessing. This procedure has to be done in order to eliminate recognition errors caused by the dissimilar line thickness. Stroke thickness is a specificity of each man's handwriting, mainly caused by different pressure put on the writing instrument. Even in the situations involving a single subject, different "writing thickness" may occur in diverse situations due to the specificities of the used pen or the writing speed. Line thickness can also arise from image acquisition and subsequent geometrical transformation (especially scale) [3].

Basically the skeleton is obtained by using recursive thinning till analyzed characters are reduced to a set of one pixels thick lines and curves. Subparts connectivity points and stroke's end pixels have to be preserved.

The straightforward approach to realize this task is employing the set of structuring elements described in (1.1)

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \ldots \qquad (1.1.)$$

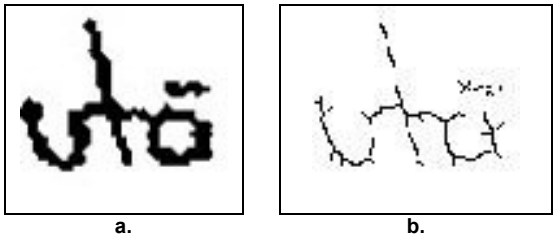
The process is a little slow but gives adequate response in the case of simple test images as the one in Fig.1.1. Because of the typographical quality requirements all images presented in this paper are negatives of the real ones.



a. **b.**

Fig.1.1.a. Test image, rectangle
b. Skeleton

Unfortunately when real life images (i.e. hand printed & handwritten documents' images) are processed the resulting skeleton is very coarse (see Fig. 1.2.). Thus a supplemental curve relaxation procedure became compulsory [4].



a. **b.**

Fig.1.2.a. Handwritten document image
b. Skeleton with recursive standard thinning

Better techniques rely on a sequence of successive thinning with homotropic structuring elements L (1.2.) or M (1.3.) able to deliver one pixel thick connected lines and curves (skeletons approximations) in a less time consuming process. The symbol "*" is used in order to mark the position where the "white" or "black" color of the pixel is of no consequence for the local thinning decision and

[1] United Technologies Paris Branch ioan_snep@uteh.fr
[2], [3] Electronic and Telecommunication Faculty, Communications Department, Bd. V. Pârvan Timişoara 1900, dlacrama@go.ro

therefore no test is to be made about them in the software implementation.

$$\begin{bmatrix}1&1&1\\*&1&*\\0&0&0\end{bmatrix}\begin{bmatrix}*&1&1\\0&1&1\\*&0&*\end{bmatrix}\ ... \qquad (1.2.)$$

$$\begin{bmatrix}1&1&1\\*&1&*\\*&0&*\end{bmatrix}\begin{bmatrix}*&1&1\\0&1&1\\*&0&*\end{bmatrix}\ ... \qquad (1.3.)$$

The other structuring elements reported in the image processing literature (C, D, E) are crafted for some slightly different targets and behave poorly with both typed and handwritten documents images.

The recursive thinning results with the two homotropic structuring elements on the same image are presented in Fig.1.3. The software implementation of the recursive thinning with all structuring elements in this paper is the same, thus the comparisons among them is made on an objective basis.

The L structuring element commonly gives better skeletons approximation when it is employed with handwriting but it is slower. The M structuring element is always quicker but the curves are coarser and disconnection errors occur.

a. b.

Fig.1.3.a. Image processed with M structuring element
b. Image processed with L structuring element

The slow - quick discussion generally tend to become obsolete with the processing speed and the vast resources of actual computers. Therefore the L element technique is preferred in almost all cases.

The problem resides in the fact that the L structuring element does not behave quite well with the handwritten documents' processing and the resulting skeletons are still coarse and sometimes thorny.

II. THE H STRUCTURING ELEMENT

Therefore the authors tried to improve the procedure performances employing an optimized structuring element resulting from combining the standard skeleton structuring element and the L structuring element.

We started from the straightforward simplification of the standard structuring element (1.1.). Each pair of matrixes can by mathematically described by a common model as shown in (2.1.).

$$\begin{bmatrix}1&1&1\\1&1&1\\0&0&0\end{bmatrix}\begin{bmatrix}1&1&1\\1&1&0\\1&0&0\end{bmatrix}\rightarrow\begin{bmatrix}1&1&1\\1&1&*\\*&0&0\end{bmatrix} \qquad (2.1.)$$

Consequently the eight matrixes in the initial set become four as shown in (2.2.):

$$\begin{bmatrix}1&1&1\\1&1&*\\*&0&0\end{bmatrix}\begin{bmatrix}*&1&1\\0&1&1\\0&*&1\end{bmatrix}\begin{bmatrix}0&0&*\\*&1&1\\1&1&1\end{bmatrix}\begin{bmatrix}1&*&0\\1&1&0\\1&1&*\end{bmatrix} \qquad (2.2.)$$

In order to improve speed and accuracy an auxiliary set of four matrixes were borrowed from the L technique.

$$\begin{bmatrix}1&1&*\\1&1&0\\*&0&*\end{bmatrix}\begin{bmatrix}*&1&1\\0&1&1\\*&0&*\end{bmatrix}\begin{bmatrix}*&0&*\\1&1&0\\1&1&*\end{bmatrix}\begin{bmatrix}*&0&*\\0&1&1\\*&1&1\end{bmatrix} \qquad (2.3)$$

Thus the H structuring elements set is:

$$\begin{bmatrix}1&1&1\\1&1&*\\*&0&0\end{bmatrix}\begin{bmatrix}1&1&*\\1&1&0\\*&0&*\end{bmatrix}\ ... \qquad (2.4.)$$

This homomorphic structuring elements set behaves better with the hand printed & handwritten documents' images, as it will be shown in section IV. The resulting skeletons are less coarse and the number of needed processing cycles is equal or less the one in L technique.

III. SOFTWARE IMPLEMENTATION

The program performing the processing tasks has been developed in Java (JDK 1.3). It is a Window application and it is able to:

- Display the document's image;
- Execute the recursive thinning until skeleton stage is reached;
- Display a contour showing the number of cycles performed;
- Save the final image resulted after the process.

The class structure contains a main class W (extending the javax.swing.JFrame and implementing the Runnable interface) and a

processing Sb class (extending java.awt.image. ImageFilter).

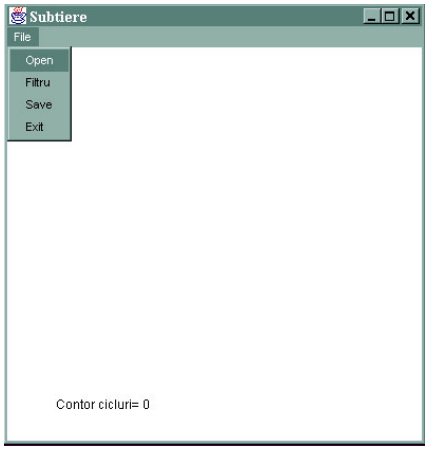


Fig.2.1. The Skeleton program frame in the starting state

The main class is the program main thread and contains code to implement the window frame. This window has a file menu with open, filter, save and exit menu items and a display image place as shown in Figure 2.1. Consequently "W" class also has the methods for performing opening, saving and exiting.

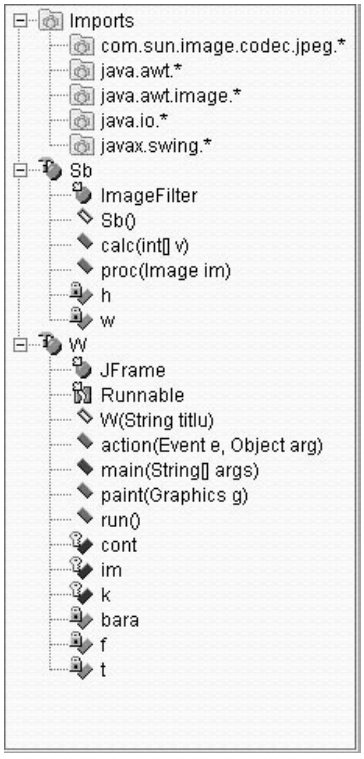


Fig.2.2. The program class structure as shown in Borland JBuilder 4 environement,s Structure Window

The Sb class is focused on the thinning and contains the set of "if...else if" statements implementing the Hit or miss procedures needed for finding the pixels to be changed from letter pixels to background ones.

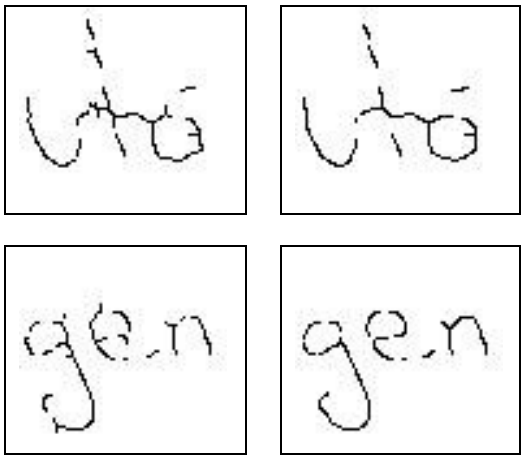
The sequence of successive thinning is realized through the Thread mechanism inside a "while" statement. It stops when Sb report no more changes has been done over the last thinning cycle.

The contour is also declared incremented and displayed within the "W" class, shows the number of cycles in real time. In the end its figure represents the skeleton stage thinning cycles needed for the currently processed image.

Final image saving has been done using the method in [2]. It gave the possibility to preserve in JPEG format the skeletonized images for the further processing steps.

IV EXPERIMENTAL RESULTS

The experiments were carried out over the standard NIST database for hand printed & handwritten characters. The processing with each set of matrixes was performed in identical software environments.



a. b.

Fig.3.1.a. Image processed with L structuring element
b. Image processed with H structuring element

Each of the four skeleton techniques were analyzed and it resulted that:

A. The standard thinning and the M homotropic structuring element technique are not able to perform satisfactory with the hand printed & handwritten documents' images.

B. The L homotropic structuring element technique behaves better, but the resulting skeleton is still coarse and it has a thorny aspect. The coarse parts and the thorns are sometimes source of inadequate final classifications.

C. The proposed H homotropic structuring element technique is more precise while it is at least as fast as the L technique. The skeletons

have almost 4% less pixels due to the absence of the coarse parts and thorns.

It is also important to stress here that the final recognition results on the experimental test set improved with 0.62% because of using H instead L homotropic technique.

The explanation of this resides mainly in the less coarse strokes that leave less space for miss interpretations.

V CONCLUSIONS

The skeletonization is an important step in preprocessing documents' images. If the skeleton is accurate it avoids a great number of mistakes in the final character recognition stage. The process consists of a recursive thinning that usually employs a homotropic set of structuring elements.

Experiments show that the largely known and used L structuring element does not behave well with the hand printed & handwritten character documents' images.

Thus the authors proposed another solution coming out from combining standard thinning and L set. The results prove to be better and the improvements also echoed in the final recognition stage.

As researches about skeleton reach the final conclusive state it became clear the need for implementing also a more flexible binarization technique in order to reduce the strokes and curves disruptions.

Team thought about reevaluating the results after a better binarization method will be find and adapted to the documents' images specificities. It is highly probable that the two preprocessing steps could be improved not only as separate entities, but also as interacting stages of the image preparation for the later automatic interpretation93.

REFERENCES

[1] Sonka M, Hlavac V Boyle R., "Image Processing, Analysis and Machine Vision", Chapman & Hall, London 1993

[2] D.L. Lacrămă, I. Şnep, "Manipulation of JPEG Images in Java", Proceedings of the AEG Symposium, Prague, 26-27 October 2001, pp 308-311

[3] Shapiro V., Gluchev G., Sgurev V., "Handwritten Document Image Segmentation and Analysis", Pattern Recognition Letters, North Holland, vol.14, No.1, January 1993, pp.71-78.

[4] Deseilligny M.P., Stamon G., Suen C.Y., "Veinerization: A New Shape Description for Flexible Skeletonization", IEEE Trans. on PAMI, vol.20, No.5, May 1998.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Une méthode nouvelle pour la compression de la parole

Andrei Cubiţchi[1]

Resumé – On présente une méthode nouvelle pour la compression de la parole qui est meilleure du point de vue du facteur de compression et de la transparence que la méthode de compression utilisée en GSM. Cette nouvelle méthode de compression utilise une transformée en paquets de cosinus adaptative. Elle réalise la séquence d'opérations suivante : le calcul de la transformé en paquets de cosinus du signal à comprimer, la sélection adaptative des plus importants coefficients obtenus et la quantification adaptative des coefficients obtenus ainsi. Pour la reconstruction du signal de départ en utilisant sa version comprimée ont été considérés des opérations inverses. Bien sur la reconstruction n'est pas parfaite mais les distorsions introduites par la compression sont négligeables, donc la méthode considérée est transparente. Le but des opérations adaptatives est de maximiser le facteur de compression en gardant sous contrôle les distorsions. On présente aussi quelques résultats de simulation.
Cuvinte cheie: parole, compression, paquets de cosinus

I. INTRODUCTION

La compression des données est utilisée souvent pour le stockage et pour la transmission des données. Il y a une grande diversité d'algorithmes de compression. Il y a deux grandes catégories, [1], les algorithmes pour la compression sans pertes et les algorithmes pour la compression à pertes. Tous ces algorithmes ont le même but, réduire la redondance de la suite de données considérée, mais ceux appartenant à la première classe réalisent une reconstuction parfaite et ceux qui appartient à la deuxième classe ne réalisent pas une telle reconstruction. Cette deuxième classe contient divers algorithmes pour la compression de la parole, de la musique ou des images. Dans ce travail on présente une méthode pour la compression/décompression de la parole appartenant à cette deuxième classe qui exploite une transformée orthogonale très connue, appelée Transformée en paquets de cosinus, TPC, [2], [3], [4], [5]. Cette transformée a été sélectionnée pour sa vitesse de calcul (comparable à la vitesse de la Transformée de Fourier rapide) et pour son adéquation au modèle de la parole. La structure de ce travail est la suivante. Dans le paragraphe numéro 2 est présenté le système de compression proposé. Le sujet du troisième paragraphe est la TPC. Le paragraphe 4 est dédié à la présentation d'une méthode de quantification très simple. On peut trouver quelques résultats de simulation dans le paragraphe suivant. Quelques conclusions sont présentées dans le dernier paragraphe.

II. LE SYSTEME DE COMPRESSION

Le système de compression proposé est présenté à la figure 1.

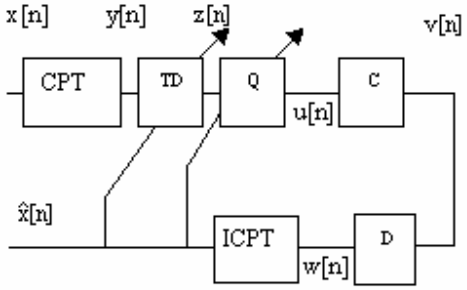


Figure 1. Le système de compression proposé.

Le premier bloc, CPT, calcule la transformée en paquets de cosinus du signal d'entrée, $x[n]$, $y[n]$. Le détecteur de seuil, TD, réalise l'opération:

$$z[n]=\begin{cases} y[n] , & si \quad |y[n]| > t, \\ 0 & si \quad non \end{cases} \qquad (1)$$

où t est la valeur d'un seuil. Q représente un quantificateur adaptatif. Le codeur C réalise une compression sans pertes supplémentaire. La sortie du système de compression est décrite par le signal $v[n]$. Le décodeur D est le système inverse du codeur C. A la sortie du système qui calcule la

[1] Goldstern, Timişoara

transformée en paquets de cosinus inverse, ICPT, on obtient le signal reconstruit, $\hat{x}[n]$. La qualité de la reconstruction peut être appréciée à l'aide du rapport signal/bruit de ce signal, snr_0 .L'erreur de reconstruction représente la distorsion introduite par la méthode de compression et peut être regardée comme une perturbation additive du signal d'entrée. Sa puissance représente la puissance du bruit dans l'expression du rapport signal/bruit. L'autre paramètre important du système de compression de la figure 1 est le facteur de compression, f_c . Celui-ci représente le rapport entre les nombres de bits des signaux $x[n]$ et $v[n]$. Le détecteur adaptatif de seuil et le quantificateur adaptatif de la figure 1 sont conçus pour maximiser f_c en gardant snr_0 supérieur à 20 dB.

III. L'UTILISATION DE LA TPC POUR LA COMPRESSION DE LA PAROLE

Il y a un grand nombre de transformées qui peuvent être utilisées pour la compression des données. Le choix d'une transformée particulière doit être fait en accord avec la spécificité du signal à comprimer. Ce travail est concentré sur les transformées en ondelettes orthogonales utilisées pour la compression de la parole. Le signal de parole a un modèle mathématique qui fait l'utilisation des paquets de cosinus très utile pour les applications de compression.

A. Le modèle mathématique de la parole

Chaque phrase est une séquence de tons qui ont différentes intensités, fréquences et durées. Chaque ton est un signal sinusoïdal avec amplitude, fréquence et durée spécifiques. C'est le modèle sinusoïdal de la parole. Une description mathématique pour ce modèle est:

$$x(t) = \sum_{q=1}^{Q(t)} A_q \cos\theta_q(t) \qquad (2)$$

[6], où les composantes sont nommées partiales. Chaque terme de cette somme est un signal à double modulation. Donc il ne s'agit pas de signaux stationnaires. Mais fréquemment la parole est considérée comme une succession de signaux stationnaires. Divisant le signal de parole en une succession de segments chaq'un ayant une durée inférieure à 25 ms, on obtient une séquence de signaux stationnaires. Sur chaque segment le modèle de la parole peut être de la forme:

$$x_s(t) = \sum_{q=1}^{Q} A_q \cos\omega_q t \qquad (3)$$

Cette décomposition ressemble beaucoup à la décomposition du signal $x_s(t)$ dans un paquet de cosinus, [2]. La décomposition du même signal, en utilisant une base d'ondelettes est de la forme:

$$x_s(t) = \sum_{k=1}^{K}\sum_{l=1}^{L} \langle x_s(t), \psi_{k,l}(t)\rangle \psi_{k,l}(t) \qquad (4)$$

où $\psi_{k,l}(t)$ sont les ondelettes générées par la mère des ondelettes $\psi(t)$. Le facteur de compression obtenu utilisant une ondelette mère spécifiée est plus grand si le nombre de coefficients non-nuls:

$$d_{k,l} = \langle x_s(t), \psi_{k,l}(t)\rangle \qquad (5)$$

de cette décomposition, N_ψ , est plus petit. Mais:

$$d_{k,l} = \int_{-\infty}^{\infty} x_s(t)\psi^*_{k,l}(t)dt = R_{x_s,\psi_{k,l}}(0) \qquad (6)$$

Donc ces coefficients représentent l'intercorrelation des fonctions $x_s(t)$ et $\psi_{k,l}(t)$ calculée en origine. C'est la plus grande valeur de cette fonction. Cette fonction mesure la ressemblance entre ces deux fonctions. Donc la magnitude des coefficients $d_{k,l}$ est plus grande si les signaux $x_s(t)$ et $\psi_{k,l}(t)$ sont plus ressemblants. En utilisant la relation (3) on peut dire que les ondelettes les plus ressemblantes au signal $x_s(t)$ sont les éléments des paquets de cosinus. Mais si l'ensemble $\{\psi_{k,l}(t)\}_{k\in Z, l\in Z}$ est une base orthonormale alors l'énergie du signal $x_s(t)$ peut être exprimée comme:

$$E_x = \sum_{k=1}^{K}\sum_{l=1}^{L} |d_{k,l}|^2 \qquad (7)$$

Parce que l'énergie du signal $x_s(t)$ est une constante indépendante de la mère d'ondelettes sélectionnée on peut affirmer que le nombre N_ψ est plus petit si la magnitude des coefficients $d_{k,l}$ est plus grande. Donc pour la compression de la parole le meilleur choix pour la mère des ondelettes est la classe de Malvar (qui conduit aux paquets de cosinus), [4].

B. AUTRES PROPRIETES DE LA TPC

La TPC est une transformée adaptative. Elle peut être optimisée en utilisant une procédure de

recherche de la meilleure base, [2]. C'est une procédure très puissante qui peut augmenter la qualité d'une certaine méthode de traitement du signal. Il y a plusieurs fonctionnelles de coût qui peuvent être minimisées pour le choix de la meilleure base. La fonctionnelle la plus utilisée est l'entropie des coefficients $d_{k,l}$, [2]. Mais la minimisation de cette fonctionnelle ne conduit pas à la maximisation du facteur de compression. La fonctionnelle de coût dont la minimisation conduit à la minimisation du nombre N_ψ et donc à la maximisation du facteur de compression est celle qui minimise le nombre des coefficients supérieurs à un certain seuil, t, [2], N_s. En effet, en utilisant cette fonctionnelle de coût pour le choix de la meilleure base (le meilleur paquet de cosinus) on obtient un certain nombre N_ψ. En rejetant tous les coefficients inférieurs au seuil t, à l'aide du détecteur de seuil (de la figure (1)), le nombre des coefficients non-nuls à la sortie du détecteur de seuil devient égal à N_s. Mais ce nombre représente une valeur minimale. Donc en utilisant cette fonctionnelle de coût le nombre de coefficients non-nuls, à la sortie du détecteur de seuil, est minimisé. C'est le motif pour lequel on peut affirmer que le facteur de compression est maximisé quand cette fonctionnelle de coût est utilisée pour la sélection du meilleur paquet de cosinus. En augmentant la valeur du seuil t, le nombre N_s devient plus petit et la valeur du facteur de compression devient plus grande. Malheureusement la valeur du snr_0 devient plus petite aussi. Donc l'augmentation de la valeur du seuil t doit être contrôlée pour garder la transparence de la compression. C'est le motif pour lequel le détecteur de seuil de la figure 1 doit être adaptatif. Un autre paramètre de la TPC qui doit être considéré pour l'optimisation de la compression est son nombre d'itérations.

IV. LE DETECTEUR DE SEUIL

En analysant le système de la figure 1 on peut constater que la distorsion introduite par la compression a la valeur:

$$D = E\left\{\left(x[n]-\hat{x}[n]\right)^2\right\} \qquad (8)$$

où E représente l'opérateur de moyenne statistique. Parce que la TPC et son inverse la TPCI sont des transformées orthogonales la dernière relation devient:

$$D = E\left\{\left(y[n]-u[n]\right)^2\right\} \qquad (9)$$

La valeur du seuil t est choisie pour satisfaire la condition:

$$D < \alpha \cdot E_x \qquad \alpha < 1 \qquad (10)$$

où E_x représente l'énergie du signal d'entrée $x[n]$. On peut prouver la proposition suivante.

Proposition 1.

Une borne superieure de la distorsion du signal reconstruit obtenu après la compression adaptative basée sur la TPC décrite dans ce travail est $N \cdot t^2$ *ou* N *représente le nombre d'échantillons du signal d'entrée.*

Preuve.
L'erreur quadratique moyenne d'approximation du signal $y[n]$ par le signal $z[n]$ est:

$$\varepsilon_1 = E\left\{\left(y[n]-z[n]\right)^2\right\} = \sum_{k=1}^{K} y^2[n_k] \qquad (11)$$

où n_k représente la position des échantillons du signal $y[n]$ avec la valeur absolue inférieure au seuil t. Il y a K tels échantillons. Soit $o[n]$ le signal obtenu après la mise en ordre croissante des échantillons du signal $y[n]$. L'erreur quadratique moyenne devient:

$$\varepsilon_1 = \sum_{k=1}^{K} o^2[k] \le K \cdot t^2 \qquad (12)$$

On considère comme première approximation que le bloc Q réalise une quantification uniforme de pas t. En utilisant une quantification non-uniforme les résultats seront mieux. L'erreur quadratique moyenne est:

$$\varepsilon_2 = \sum_{n=1}^{N} \left(z[n]-u[n]\right)^2 \qquad (13)$$

Pour chaque échantillon de la séquence $o[k]$, $k = \overline{1,K}$ on associe un zéro dans la séquence $u[n]$. Pour les autres échantillons du signal $z[n]$ la différence $z[n]-u[n]$ est inférieure à la valeur t. C'est le motif pour lequel on peut écrire:

$$\varepsilon_2 \le \sum_{k=K+1}^{N} t^2 = \left(N-K\right)\cdot t^2 \qquad (14)$$

Parce que la distorsion définie en (9) peut être écrite dans la forme :

$$D = \varepsilon_1 + \varepsilon_2 \qquad (15)$$

tenant compte des relations (12) et (14) on peut conclure que la proposition est prouvée.
Donc pour garder la distorsion sous la valeur $\alpha \cdot E_x$ est suffisant de choisir la valeur de seuil:

$$t = \sqrt{\frac{\alpha \cdot E_x}{N}} \qquad (16)$$

La constante α peut être mise en relation avec le rapport signal/bruit des signaux $u[n]$ et $\hat{x}[n]$ (qui ont la même énergie), snr_0. Donc on peut écrire:

$$snr_0 = 10 \cdot \log_{10} \frac{E_x}{D} \geq -10 \cdot \log_{10} \alpha \qquad (17)$$

Ainsi une borne inférieure du rapport signal/bruit, qui dépende de α:

$$\beta = -10 \cdot \log_{10} \alpha \qquad (18)$$

a été établie. En prenant le signe égal dans la dernière relation on peut obtenir une borne inférieure pour la constante α:

$$\alpha_m = 10^{-\frac{\beta}{10}} \qquad (19)$$

En utilisant cette valeur et la relation (16) une borne inférieure de la valeur du seuil t peut être obtenue:

$$t_m = \sqrt{10^{-\frac{\beta}{10}} \frac{E_x}{N}} \qquad (20)$$

En conclusion en choisissant pour le seuil t une valeur superieure à t_m on obtient une valeur du rapport signal/bruit à la sortie superieure à β. Malheureusement la valeur exacte du snr_0 ne sera pas connue. C'est le motif pour lequel un algorithme adaptatif pour le choix de la valeur du seuil est recommandé. Cet algorithme peut utiliser la valeur t_m pour initialisation. En commencent avec cette valeur t est augmenté. A chaque itération la valeur du snr_0 est calculée. Si cette valeur est superieure à β le processus d'augmentation est continué. L'algorithme prends fin quand pour la première fois la valeur de snr_0 devient inférieure à β.

V. LE QUANTIFICATEUR

Les considérations faites dans le paragraphe précédent sont basées sur l'hypothèse d'une quantification uniforme. Quand on utilise un quantificateur non-uniforme on peut obtenir des résultats meilleurs. Une solution très simple pour une quantification non-uniforme a été choisie pour le système de compression proposé dans ce travail. La séquence obtenue à la sortie du détecteur de seuil de la figure 1, le signal $z[n]$, est divisée en 32 blocs, $z_k[n]$, $k = \overline{1,32}$ de même longueur. En chaque bloc on réalise une quantification uniforme. Les plus grandes valeurs des signaux $z[n]$, z_M et $z_k[n]$, z_{kM} sont détectées. Pour chaque bloc est alloué un certain nombre de bits. Cette procédure est basée sur les valeurs z_{kM}. Pour la valeur z_M sont alloués 6 bits (2^6 niveaux de quantification). Pour les valeurs z_{kM} sont alloués:

$$\gamma_k = \left[\left[\frac{z_{kM}}{z_M} \cdot 2^6 \right]\right] \qquad (21)$$

niveaux de quantification, où $[[\ ]]$ représente la partie entière.
Ainsi un nombre de $b_k = [[\log_2(\gamma_k + 1)]]$ bits sont alloués pour chaque échantillon du bloc numéro k. La quantification du bloc numéro k est réalisée en utilisant la transformation:

$$u_k[n] = \left[\left[\frac{z_k[n]}{z_{kM} + 0.01} \cdot \gamma_k \right]\right] \qquad (22)$$

Ainsi une normalisation de niveau est réalisée en chaque bloc. La denormalisation correspondante doit être réalisée pendant la phase de reconstruction, avant le calcul de la TPCI. Cette opération est attribuée au bloc D de la figure 1. Le grand avantage de la procédure de quantification proposée vient de la propriété de decorrelation de la TPC. Grâce à cette propriété beaucoup des valeurs z_{kM} sont nulles. Les valeurs correspondantes γ_k et b_k sont nulles aussi. Donc le nombre total de bits alloués aux échantillons du signal $u[n]$, N_b, est très petit par rapport au nombre de bits du signal $x[n]$. Cette procédure de quantification a un petit désavantage aussi. Pour la transmission ou pour le stockage de chaque bloc

du signal $u[n]$, $u_k[n]$, il faut ajouter aux coordonnées de chaque échantillon quelques valeurs supplémentaires, les valeurs z_{kM}. En utilisant ces valeurs et la relation (21) les nombres γ_k peuvent être calculés aussi pendant la phase de reconstruction. A l'aide des paramètres z_{kM} et γ_k les opérations inverses aux opérations décrites par la relation (22) peuvent être réalisées. Mais le nombre de bits demandés pour la représentation des valeurs z_{kM} est très petit par rapport à N_b. Donc le facteur de compression n'est pas affecté par la nécessité d'ajouter les valeurs supplémentaires pour chaque bloc $u_k[n]$. La valeur du facteur de compression réalisé par le système de la figure 1 peut être calculé en utilisant la relation :

$$f_c = \frac{16 \cdot N}{N_c + N_p + B} \qquad (23)$$

où N_p représente le nombre de bits demandés pour le codage des positions des échantillons du signal $u[n]$ et B représente le nombre de bits demandés pour le codage des paramètres. Les nombres N_c et B peuvent être calculés en utilisant les relations suivantes:

$$N_c = \sum_{k=1}^{32} N(k) \cdot b_k \qquad (24)$$

où $N(k)$ représente le nombre d'échantillons non-nuls dans le bloc numéro k et:

$$B = \sum_{k=1}^{32} b_k \qquad (25)$$

Le nombre N_p peut être calculé en utilisant la relation:

$$N_p = \sum_{k=1}^{32} N(k) \cdot \zeta_k \qquad (26)$$

où ζ_k représente le nombre de bits demandés pour la représentation de la position de chaque échantillon de valeur non-nulle du bloc numéro k. Parce qu'en chaque bloc il y a un nombre maximal de 32 telles positions les valeurs de ζ_k sont inférieurs à 5. Donc une borne superieure pour N_p est:

$$5 \cdot \sum_{k=1}^{32} N(k) = 32 \cdot N_n \qquad (27)$$

VI. LES AUTRES BLOCS DU SYSTEME DE COMPRESSION

L'utilisation du codeur C de la figure 1 augmente le facteur de compression. Ce système réalise une compression sans pertes. C'est le motif pour lequel son utilisation n'affecte pas la valeur du snr_0. L'implémentation de ce bloc fait appel à une technique traditionnelle de codage comme par exemple le codage de Huffman ou le codage arithmétique, [1]. Le signal $v[n]$ obtenu à la sortie du codeur représente le résultat de la procédure de compression. C'est le signal qui est stocké ou transmis. Les autres deux blocs du système de la figure 1 sont utilisés dans la phase de reconstruction. Le bloc D réalise le décodage du signal $v[n]$. On obtient à sa sortie les signaux $u_k[n]$ et la séquence z_{kM}, $k = \overline{1,32}$. En utilisant la relation (21) on calcule les valeurs γ_k. La denormalisation décrite par la relation:

$$w_k[n] = \frac{u_k[n]}{\gamma_k + 0.01} \cdot z_{kM} \qquad (28)$$

est réalisée. Par le concatenage des blocs $w_k[n]$ on obtient le signal $w[n]$. Le dernier bloc de la figure 1 calcule la TPCI. Le résultat est le signal $\hat{x}[n]$. C'est le résultat de la procédure de reconstruction. En utilisant ce signal on peut calculer la distorsion D. Toutes les opérations déjà décrites sont répétées, pour différentes valeurs du seuil t pour la maximisation du f_c, avec la contrainte que snr_0 soit supérieur à β.

VII. RESULTATS DE SIMULATION

Le système décrit plus haut a été simulé en Matlab. Un morceau de parole a été segmenté en blocs de 1024 échantillons. Pour chaque segment la procédure présentée a été appliquée.

La TPC a été calculée en utilisant une ou deux itérations. Il y a deux types de segments: les segments tonals et segments de bruit (où la forme d'onde ressemble à celle d'un bruit).

La valeur moyenne pour le facteur de compression de ces 20 segments est 11,54. C'est le motif pour lequel on peut dire que la méthode de compression proposée ici est superieure à la méthode utilisée en GSM. Le plus petit rapport signal/bruit enregistré sur ces 20 segments est de 18,49 dB. Le facteur de compression peut être augmenté sans perdre la qualité si chaque TPC d'un segment tonal est calculée en utilisant une seule itération. Une autre possibilité d'augmenter le facteur de compression est d'utiliser des

segments plus longs (par exemple contenant 2048 échantillons).

VIII. CONCLUSIONS

Dans ce travail on a été prouvée l'efficience de la TPC pour la compression de la parole. Dans [9] est prouvée la convergence asymptotique de cette transformation vers la transformée de Karhunen-Loeve (la meilleure transformée pour la compression d'un signal).
Un nouvel algorithme de compression a été présenté. Cet algorithme est caracterisé par l'utilisation d'un systeme de detection de seuil adaptatif. Celui-ci realise le choix du seuil en gardant l'errure quadratique moyenne de reconstruction sous une valeur imposée. L'algorithme de selection du seuil peut etre initialisé en utilisant la valeur de la relation (20). La methode de compression decrite se base aussi sur l'utilisation d'un systeme de quantification adaptatif. A la méthode de compression présentée il faut ajouter une procédure de codage des valeurs et positions des échantillons obtenues à la sortie du quantificateur. Cette procédure n'est pas encore écrite mais la méthode de compression proposée semble être très utile parce que sans cette procédure de codage la nouvelle méthode est déjà superieure à la méthode de compression utilisée dans le standard GSM.

IX. REFERENCES

[1] N. Moreau. Techniques de compression des signaux. Masson, 1995.
[2] M. V. Wickerhauser. Adapted Wavelet Analysis from Theory to Software. A. K. Peters Wesley, 1994.
[3] Y. Meyer. Ondelettes et algorithmes concurrents. Herman, Paris 1993.
[4] H. S. Malvar. Lapped Transforms for Efficient Transform/Subband Coding. IEEE Trans. on ASSP, vol.38, pp. 969-978, June 1990.
[5] M. V. Wickerhauser. Acoustic signal compression with wavelet packets, in "Wavelets-A tutorial in theory and applications", (C.K. Chui, editor), Academic Press, 1992, pp. 679-700.
[6] R. Boite, M. Kunt, Traitement de la parole, Presses Polytechniques Romandes, Lausanne, 1987.
[7] Eva Wesfreid, M. V. Wickerhauser. Etude des signaux vocaux par ondelettes de Malvar, Quatorzième colloque GRETSI, Juan-les-Pins, 1993, pp. 379-382.
[8] T. Asztalos, A. Isar, "Wavelets and Audio Data Compression", IEEE International Conference on Signal Circuits and Systems, SCS'99, 5-7 July, 1999, Iasi, Romania.
[9] A. Cubitchi, "Metodă de compresie a semnalului vocal bazată pe utilizarea funcțiilor "wavelet"", Referat nr. 3, în cadrul pregătirii pentru doctorat, conducător ştiințific Prof. dr. ing. I. Naforniță, Timişoara, 2001.

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Romanian Bank-Notes Recognition Using Cellular Neural Networks

Corina Botoca[1]

Abstract - It is presented a new method to avoid bank-notes counterfeiting on color copiers. This method use a sequence of analogic algorithms on cellular neural networks (CNN) to perform complex image processing.

Keys words: cellular neural network, analogic algorithm, CNN template

I. INTRODUCTION

A CNN is a nonlinear (piecewise), analogue, multidimensional processing array [1]. Its dynamics is described by the following equations:

$$\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{kl \in N_r} A_{ij,kl} \cdot y_{kl}(t) + \sum_{kl \in N_r} B_{ij,kl} \cdot u_{kl}(t) + I_{ij} \quad \textbf{(1)}$$

$$y_{ij}(t) = \frac{1}{2}\left(|x_{ij}(t)+1| - |x_{ij}(t)-1|\right) \quad \textbf{(2)}$$

where -x_{ij} is the state of the C_{ij} cell;
-y_{ij} is the output of the C_{ij} cell;
-u_{kl} is the independent input in the C_{kl} cell;
-$A_{ij,kl}$ is the weight of the feedback connection; from the C_{kl} cell to the C_{ij} cell;
-$B_{ij,kl}$ is the weight of the control connection from the C_{kl} cell to the C_{ij} cell;
-I_{ij} is the bias current of the C_{ij} cell;
-N_r is the neighborhood of C_{ij} cell;

The CNN most important characteristics are their geometrical and electrical regular structure, the locality of interconnections between the processing elements and their programmability. These characteristics permit to a CNN to solve complex image processing tasks through the implementation of analogic algorithms, that can be combined according to a flowchart,. Some of the applications are: interpolation and three-dimensional image reconstruction [3], image compression and decompression [6], microscopic and topographic image processing [2],[4], character recognition , Canadian bank-notes recognition[5] .

In this paper it is presented a method to stop the operation of a color copy-machine whenever is detected a fraud tentative, using different analogic algorithms and local logic of CNN cell. Similar algorithms are presented in [5] and used to recognize the Canadian dollars. The method was developed by the author in Matlab considering the particularities of Romanian banknotes.

II.BASIC CONCEPTS

The most important aspect in recognizing a bank-note is to select the significant features of the tested image. These features can be : chromatic, topologic or related to the objects size. For instance, the characteristics selected to recognize the 10.000 bank note are presented in Fig.1.

The analogic CNN algorithm has to test if the selected feature is present or not in the image. The CNN algorithm is similar to a sequential classic subroutine. Its output image is binary . The image pixels have a value of +1 (true) in locations where the feature was found and –1 (false) in the rest of the image.

Fig.1

Results of different algorithms can be combined applying spatial logic CNN operators. All these operations can be processed on chip without read/write , except original input and final output.

Another important aspect of the problem is the processing speed. The modern copy-machines must have a very good resolution and a copying speed greater than four or five copies/second. So the time

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii Bd. V. Pârvan Timişoara 1900 e-mail corinab@hermes.ee.utt.ro

necessary to verify the document must not exceed 200 ms. Due to their parallel processing and possible hard implementation the CNN provide an efficient solution.

Recognizing bank-notes has to satisfy two requirements:

-to avoid counterfeiting bank-notes, in any orientation of the image or background;

-to permit copying any other documents;

III.THE RECOGNITION PROGRAM

The recognition program consists of many algorithms, according to the flow chart from Fig.2:

- color filtering that generates the set of suspected objects from the image;
- size classification ;
- objects containing more than one hole extraction;
- objects containing more concavities extraction;
- objects with one hole extraction;
- reconstruction of the extracted objects;

Each module from Fig.2 is in fact a subroutine performed by a sequence of elementary locally operations, named templates. Some of these subroutines will be presented in the following.

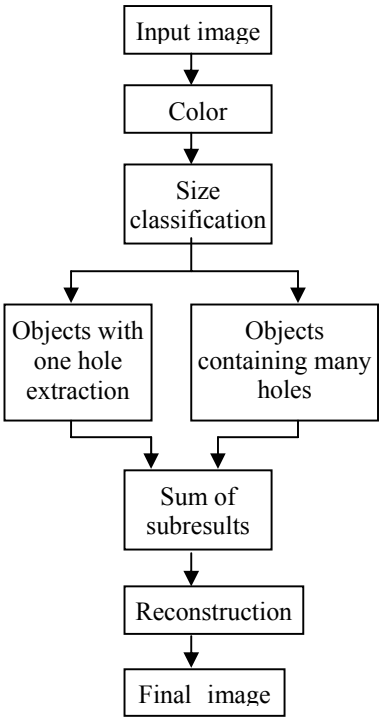


Fig2 The flow chart of the recognition program for the 10.000 lei banknote

The color filtering subroutine flow chart is illustrated in Fig.3. Each pixel of the image has a value related to its color in –1 (for white) and +1(for black) interval. The color filtering selects all the objects that have colors values between an interval of

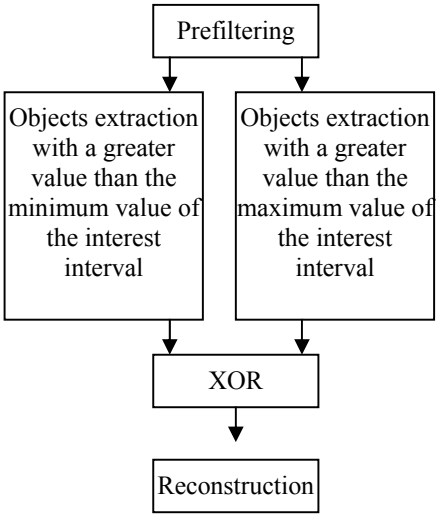

Prefiltering
Objects extraction with a greater value than the minimum value of the interest interval
Objects extraction with a greater value than the maximum value of the interest interval
XOR
Reconstruction

Fig.3 The color filtering flow chart

In the second step there are converted in black all the objects that have a greater value than the minimum value of the interest interval of colors and all the objects that have a greater value than the maximum value of the interest interval of colors. In the third step it is processed a XOR operation between the images resulted in the second step. Finally there are reconstructed all the objects that are in the desired interval.

interest and eliminates all the others. In the first step is performed pre-filtering of the image.

In Fig 4 there are presented the effects of color filtering on the 10000 lei banknote. Size classification subroutine selects from the output image of color filtering all the objects that have a dimension between an interval of pixels and deletes all the other objects.

The steps of the size classification subroutine are illustrated in Fig.5. From its output image the subroutines ″objects containing many holes extraction″ and ″objects with one hole extraction″ will eliminate all the objects that have more than one hole and other uninteresting characteristics and will retain only objects with one hole. The two images obtained at the output will be applied as image a) and b) to the reconstruction subroutine as it can be seen in Fig.6.

The output image will contain the group of selected objects from Fig.1. In this situation the CNN will stop the banknote copying.

IV. THE RECONSTRUCTION TEMPLATE

As example of an elementary operation it will be presented the reconstruction template, because most of the subroutines uses it. This operator extracts from a set of objects the one with a desired feature (signature). It uses two images. The image containing many objects is applied as input to the CNN. The

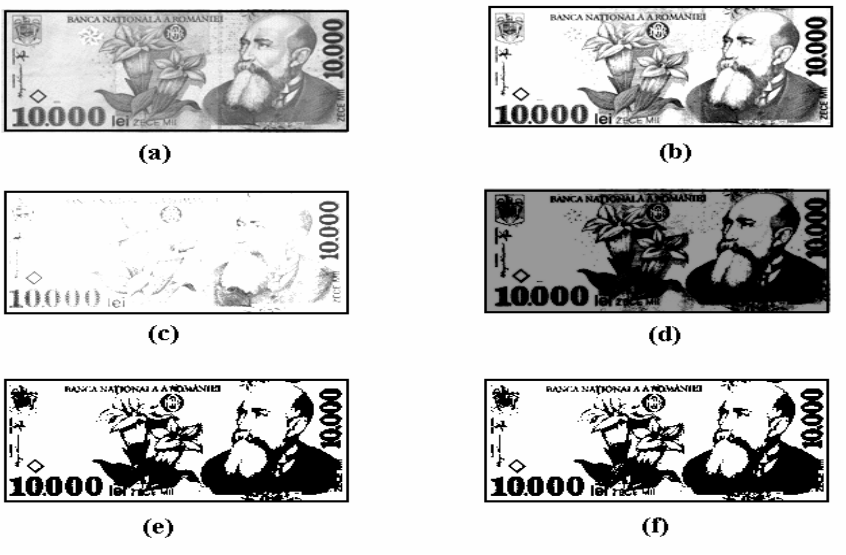


Fig.4. The steps of color filtering a) input image; b) prefiltering result; c) Objects with a greater value than the minimum value color of interest; d) Objects with a greater value than the maximum value color of interest; e)XOR between image c) and d); f) output image

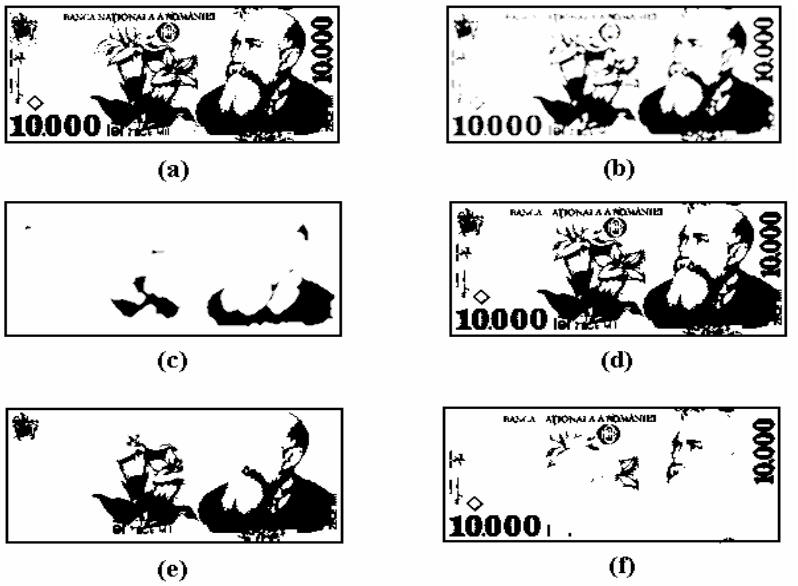


Fig.5 Steps of size classification: a) input image; b) image with n pixels levels deleted; c) image with m-n pixels levels deleted; d) reconstruction of the image b); e) reconstruction of the image c); f) XOR application to d) and e)

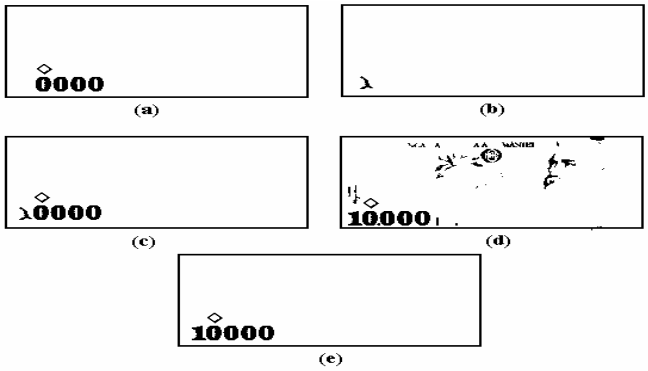


Fig.6

image containing the signature is applied as CNN state. When the transition settles down at the output of the CNN will be obtained the reconstructed object with a particular signature, as it can be seen in Fig.7 The reconstruction template is the following:

$$A=\begin{bmatrix}0.5 & 0.5 & 0.5\\ 0.5 & 4 & 0.5\\ 0.5 & 0.5 & 0.5\end{bmatrix},\quad B=\begin{bmatrix}0 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 0\end{bmatrix},\quad I=2.5$$

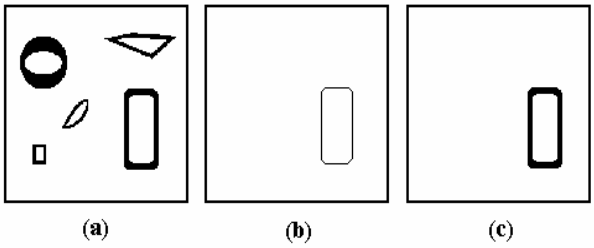


Fig.7 Object reconstruction based on signature: (a) the original image; (b) the signature; (c) reconstructed object

The processing time increases proportional with the size of the reconstructed object.

V. EXPERIMENTAL RESULTS

The recognition program was implemented in MATCNN, a toolbox of Matlab for CNN. It was tested on different Romanian banknotes: on the 5000, 10.000,50.000 and 100000 lei banknote with different backgrounds. In some situations it was enough to apply only some of the subroutines from Fig.2 to obtain the recognition characteristics. If the presence of an banknote was confirmed in the input image, the recognition characteristics (see example from Fig.1) were generated at the output. So a copy-machine equipped with a a CNN will stop functioning whenever it detects a fraud tentative.

VI. CONCLUSIONS

The significant features of Romanian banknotes were selected, it is presented a method that use the local logic of CNN and developed a program in Matlab capable to prevent them to be copied by color copy-machine. The algorithm benefit from the high speed of CNN and a possible hard implementation , that can offer a real time operation.

VII. REFERENCES

[1] L.O.Chua , .L.Yang, "Cellular Neural Networks. - Theory" IEEE Transactions on Circuits and Systems, vol 36, nr.10, octombrie, pag.1257-1272, 1988

[2] J.Miller, T.Roska, T.Szirányi, K.Crounse, L.Chua, L.Nemes, "Deblurring of images by Cellular Neural Networks with Applications to Microscopy", The Third IEEE International Workshop on Cellular Neural Networks and Their Applications, Roma, Proceedings,199

[3] L.Nemes, G.Tóth, T.Roska, A.Radványi, "Analogic CNNAlgorithms for 3D Interpolation, Aproximation and Object Rotation using Controlled Switched Templates" , MTA-SZTAKI DNS-1994

[4] Á.Zarándy, T.Roska, Gy.Lisyka, J.Hegyesi, L.Kék, Cs.Rekeczky, "Design of Analogic Algorithms for Mammogram Analysis", IEEE Third International Workshop on Cellular Neural Networks and Their Applications,Roma,1994,Proceedings

[5] Á.Zarándy, F.Werblin, T.Roska, L.Chua, "Novel Types of Analogic CNN Algorithms for Recognizing Bank-notes" , IEEE Third International Workshop on Cellular Roma,1994,Proceedings

[6] P.Venianter, F.Werblin, T.Roska, L.Chua, "Analogic CNN Algorithms for Some Image Compression and Restoration Tasks, UCB-ERL Memo, Berkeley, 1994

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 1, 2001

Evaluating Objective Audio Quality for Minidisc and Influence of Multiple Generations Audio Coding for Perceptual Quality

Konrad V. Pfaff[1]

Rezumat - Datorită răspândirii echipamentelor ce folosesc tehnici de codare audio cu debit binar redus (dbr) în activitatea studiourilor de radiodifuziune, de la contribuţie (inclusiv înregistrările primare, din teren), prin studioul de producţie până la distribuţie şi emisie, a apărut problema codărilor multiple, succesive sau în "tandem" cum mai sunt denumite. Într-un lanţ digital de transmitere a materialelor audio pot exista combinaţii diferite de scheme perceptuale de codare succesive. S-a observat că în funcţie de algoritmii şi debitele binare folosite calitatea finală a sunetului poate fi serios afectată. În partea experimentală a lucrării a fost evaluată obiectiv calitatea audio a codării pentru algoritmul ATRAC, minidiscul fiind în prezent principalul instrument pentru înregistrările pe teren.

I. INTRODUCTION

The central objective in audio coding or audio compression is to represent a digital audio signal with a minimum of bits per sample while keeping transparent (not distinguishable from the original under given listening conditions).

The rapid acceptance and deployment of low bitrate audio coding systems in the broadcast transmission network has raised new issues with respect to maintaining the audio quality of the original program signal.

Tandem coding is one of the most important issues related to audio performance in the low bitrate digital network. Quality degradation due to transcoding between different formats can be avoided by adhering to a common system and by following the ITU-R (International Telecommunications Union, Radiocommunications sector) recommended bitrates throughout the network transmission chain.

The motivation for this paper was derived from reports, criticizing the sound quality of audio codecs within tandem configurations. While systems incorporating low bit-rate audio coding will exhibit proper performance when tested alone, the sound quality may significantly decrease along with multiple generations coding. The fact that a perceptual codec cannot be assessed by traditional audio measurements (e.g. SNR, THD+N) pinpoints another problem: The performance of codecs within tandem configurations must be evaluated by subjective listening tests.

Degradation of the sound quality caused by cascaded audio channels is well known from analog recording. Each step of copying will introduce additional tape noise, linear and non-linear distorsions. Up to a certain level (where no bit errors occur) digital audio is free of such "analog domain" artefacts. By introducing perceptual coding systems, however, due to quantization processes a different situation arises.

Fig.1 shows the basic signal path through a cascade of n audio coding devices. Every codec employs a coding algorithm (referred to as A,B, ...) at a specific bit-rate (b1, b2, ...). Because multiple generations audio coding can employ one or more types of codecs operating at identical or different bit-rates, there are defined the following classification for tandem conditions [2]:

Class A: Homogeneous Tandems

One particular coding algorithm will be applied

A1: at an identical bit-rate (e.g. copies from/to digital audio recorders, like MD-to-MD)

A2: at different bit-rates (e.g. DAB network with ISO/MPEG Layer2: contribution at 180kbit/s/ch – distribution at 180kbit/s/ch – emission at 128kbit/s/ch)

Class B: Heterogeneous Tandems

Different coding algorithms will be applied

B1: at an identical bit-rate (e.g. ISDN transmission with ISO/MPEG Layer2 and Layer3, both at 64kbit/s/ch)

B2: at different bit-rates (e.g. ISDN transmission with ISO/MPEG Layer3 at 64kbit/s/ch – harddisk recorder with Layer2 – MD)

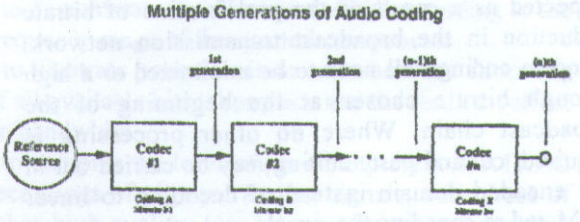


Fig. 1 Basic principle of multiple generations audio coding

[1] Chief Engineer of Radio Timişoara, a regional studio of Romanian Broadcasting Society, e-mail: kpfaff@radiotimisoara.ro

II. OPERATIONAL PRACTICES

With increasing deployment of low bitrate audio coding, use of audio compression can happen at various stages (contribution to production/studio, distribution between production facilities, emission/transmission of content to consumers etc.). Since processing/transmission of audio is still mostly done in uncompressed or even analog representation, these leads to repeated cycle of decoding, processing and re-encoding of the audio content. Similarly, a change of audio coding formats and bitrates usualy requires a decoding/re-encoding cycle for format conversion.

The introduction of MPEG coding into broadcasting brings benefits and problems. We are able to do things that were previously impossible, but there are things that we used to do that are made more difficult. The case where equipment using MPEG compression has to be integrated with existing, conventional equipment presents one kind of problem to be overcome. The other problem is to have equipment that can perform processes on MPEG compressed signals that would normally be associated with processing in the uncompressed domain.

It is the conversion from MPEG format into uncompressed signals that introduces impairements into our signals. Keeping them in MPEG-coded form preserves the quality. This is similar to the situation with analog and digital signals. Once converted to digital, it is conversions back to analog that introduce impairement. Keeping the signals in digital form preserves the quality, provided that we can perform in the digital domain all the operations to which we are accustomed in the analog domain [3].

When coding previously coded audio using modern psycho-acoustic-based coders, the audio quality deteriorates. The reasons include: the packet or frame structure imposed on the audio, the time-to-frequency transformation applied to the signal and the number of bits allocated to each part of the spectrum, which is determined by a psychoacoustic model.

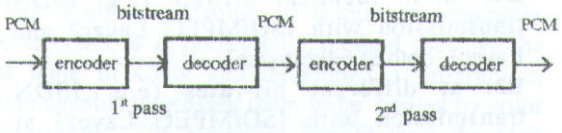


Fig.2 Diagram of cascading

Many changes in operational practices can be expected as a result of the proliferation of bitrate reduction in the broadcast transmission network. Tandem coding will need to be minimized or a high enough bitrate chosen at the beginning of the broadcast chain. Where no other processing is required, cut and paste editing may be carried out in the encoded domain instead of decoding to linear PCM and re-encoding the signal.

Although a codec may be found "transparent" for one encoding/decoding pass, serious impairements can become evident after just a few tandem passes through the same codec. These impairements are often signal dependent and so a simple "spot check" is not enough to verify signal quality of a tandem chain.

Audio coding algoritms based on auditory masking attempt to shape the quantization noise spectrum to fall below the input signal spectrum (or according to psychoacoustic models), causing the noise to be masked by the signal. If a sufficient margin is not included in the "noise-to-mask" ratio, subsequent equalization may alter the masking characterisitics of the signal, rendering this noise audible.

In studio applications, multiple decoded signals will be mixed and then re-encoded for transmission. This may occur up to several times in the course of developing a broadcast program. It is conceivable that mixing two transparently decoded audio signals which have different noise masking characteristics could result in a signal in which the quantization noise become audible. Monophonic compatibility has been identified as a requirement for high quality audio coding algorithms. The mixing of left and right channels of an encoded stereo signal should cause no audible impairement of the resulting signal (other than the loss of the stereo effect).

The development of audio processing algorithms and equipment frequently requires subjective testing to evaluate the quality of the processed audio. The lack of correlation of perceived quality with classic objective audio measurement such as SNR, THD and IMD is now well known and the increasing importance of psychoacoustic-based audio processing is rendering classic measures obsolete for many applications.

Analog interface design is important since the performance of A/D and D/A conversions can be a larger factor in analog tandem performance than actual coding artifacts, depending on the bitrate used. Sufficient precision must be provided to ensure that analog noise summation is not a limiting factor.

Adaptive Transform Acoustic Coding (ATRAC) used in minidisc (MD) is one of a number of high-quality digital audio compression schemes that are based on psycho-acoustic principles. Other schemes of this nature include the Motion Picture Expert Group (MPEG) 1 Layer 3 coding scheme, commonly known as MP3. The current popularity of MP3 and the most used free player WinAmp make it the ideal candidate for a direct subjective comparison with ATRAC.

III. QUALITY CONSIDERATIONS

Audio Coding artifacts differ in their nature and their audibility much from well known audio distorsions (wow&flutter, tape saturation, crosstalk, non-linearity, group delay distorsion, intermodulation etc.) and they may be very difficult to identify in a networked audio system, being a cascade of different signal processing devices.

Common types of artifacts

Unlike analog hi-fi equipment or FM broadcasting, perceptual encoders when run at too low bit-rates or with wrong parameters exhibit sound deficiencies which are in most cases different from noise or distorsion characteristics we all are used to. The

reason for this is the process generating differences in sound. The error introduced by a high frequency resolution perceptual coder is usually best modeled by a time-varying (in the rhythm of the basic block or frame length) error at certain frequencies, but not constrained to the harmonics of the music signal.
So the signal may be sounding:
- distorted, but not like harmonic distorsions
- noisy, but with the noise introduced only in a certain frequency range
- rough
If an encoder runs out of bits, i.e. it does not find a way to encode a block of music data with the required fidelity (e.g. allowed noise per critical band) within the bounds of available bit-rate, some frequency lines might get set to zero (deleted). The most common case is that some high frequency content is lost. If the loss of the bandwidth is not constant, but changing frame to frame, the effect becomes more objectionable than in the case of a constant bandwidth reduction.
Preechoes are a very common and famous possible artifact for high frequency resolution perceptual coding systems. To understand preechoes, let us have a look at the decoder of a perceptual coding system (fig.3). The reconstructed frequency lines are combined in the synthesis filterbank. This filterbank consists of a modulation matrix and a synthesis window. The quantization error in the coder can be seen as a signal added to the original frequency line. The length (in time) of such a signal is equal to the length of the synthesis window. Thus, reconstruction errors are spread over the full window length. If the music signal contains a sudden increase in signal energy (like a castanet attack), the quantisation error is increased too. As explained above, this quantization error (noise) signal is spread over the length of the synthesis window. If the attack occured well within the analysis window, this error will precede the actual cause for its existence in time. If this prenoise extends beyond the premasking (backwards masking) period, it becomes audible and is called preecho. There are a number of techniques to avoid audible preechoes (including variable bit-rate coding, local increase in the bit-rate locally to reduce the amplitude of preecho) but overall this type of artifact belongs to the most difficult to avoid [4].

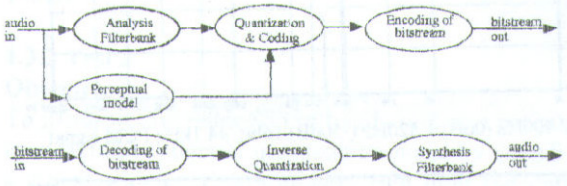


Fig. 3 Block diagram of a perceptual encoding/decoding system

Especially at low bit-rates and lower sampling frequencies there is a mismatch between time resolution of the coder and the requirements to follow the time structure of some signals. This effect is most noticeable for speech signals and for listening via headphones. The effect is sometimes called *double speak* because a single voice tends to sound like the same recording was done twice (with the same speaker) and overlayed.

How to measure codec quality
To measure codec quality of high quality audio coders there are basically three measurement methods. Listening tests, simple objective measurement methods and perceptual measurement techniques.
The ITU–R with input from a number of broadcasters and the MPEG audio group, has developed a very elaborate set of rules for listening tests. The aim of these tests is to stress the encoders under worst case conditions, to find the material which is most difficult to encode and to exhibit the performance of the encoders under test for this material. This follows the observation that in a lot of cases coding artifacts become audible only after an extensive training period.

IV. EXPERIMENTAL DETAILS

Assessment of high-quality, reduced bit-rate, perceptual audio coding schemes is not always accurate using traditional methods of electro-acoustic testing e.g. applying sweeping sine waves to measure signal to noise ratio. The nature of perceptual coding, using ATRAC as an example, implies that if the input signal to the encoder is simple in spectral content, more bits will be available for it during the allocation process. Hence the amount of audio information that is converted to quantisation noise or removed is reduced. Additionally, coding systems that use an adaptive (MDCT) transform block size, exhibit greater frequency resolution with slow transient response signals as a result of a longer transform block.
Therefore, there is a need to create test signals that can be used to assess the performance of the coding scheme over a variety of programme types. These might include basic complex waveforms with harmonic content derived of psycho-acoustics, waveforms that are chosen to illustrate deficiencies specific to the type of coding scheme being evaluated.

4.1 Experimental Aims
The objectives of the experiments executed, in order objectively to evaluate the ATRAC coding system as implemented on Sharp MD722 portable recorder, were as follows:
- to demonstrate the increased quantisation noise of spectral components as a result of ATRAC encoding, using 16bit, 44.1kHz source signals
- to demonstrate the allocation of bits according to the frequency range of the spectral components
- to illustrate potential situations where the allocation of all available bits takes place and the resultant effect on bandwidth
- to compare the level of pre-echo present in ATRAC recordings of highly transient signals, with those of other high quality, low bit rate digital audio coding systems
- to illustrate potential problems, in the time domain, which occur as a result of ATRAC encoding over multiple generations.

4.2 Experimental Procedures

- Source waveforms were digitally created using SoundForge 5.0 waveform generator, apart from those signals used for transient tests, which were taken from a test CD (AES). All waveforms were 16bit, 44.1KHz sample rate.
- The constituent spectral content of each digitally generated waveform was selected based on a theoretical understanding of how the object under test operates.
- Sine waves were used for tests based on theories of perceptual masking whilst square waves were used as a demonstration of a situation where all available bits might be used, hence resulting in the maximum amount of quantization noise that is likely to occur as a result of data reduction.
- Digital transfer for recording of all test items was employed as shown in fig.4, via the Sony Philips Digital Interface (S/PDIF) optical input of the Sharp MD722, eliminating the posibility of analogue circuitry affecting the test outcome.
- Once recorded onto MD, the test signals were then transferred back to the PC using the optical digital output (S/PDIF) of a SONY MDS-JB920 and saved as new 16bit, 44.1kHz sound files (.WAV format)
- The new sound files of the coded signals were subject to Fast Fourier Transform (FFT) analysis using SoundForge 5.0. The original source signals were also analysed for comparative measures, using the same software package.

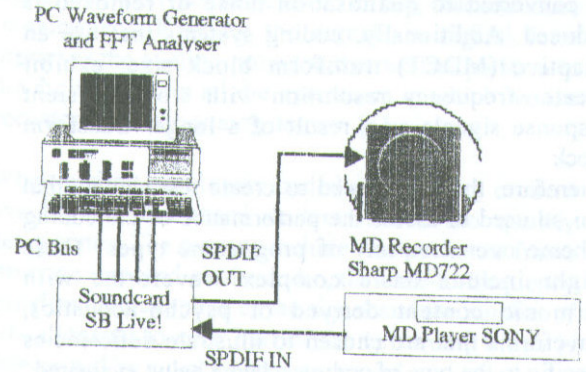


Fig.4 General set-up for transfer of objective test waveforms

The multiple generation test was made also in two more ways: a) analogue transfer to and from the Sharp MD722 to see the effect of D-A and A-D conversion the analogue circuitry on the test outcome as shown in fig.5

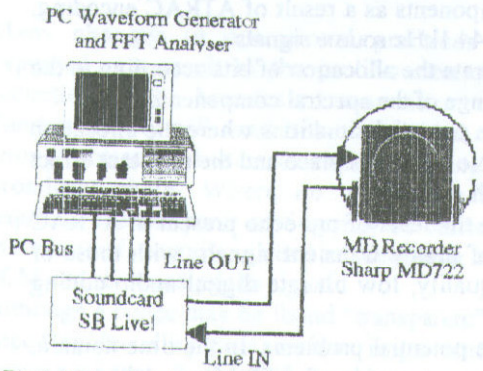


Fig.5 General set-up for analogue transfer of objective test waveforms

b) digital transfer for playback and recording using only the SONY MDS JB-920 as shown in fig.6.

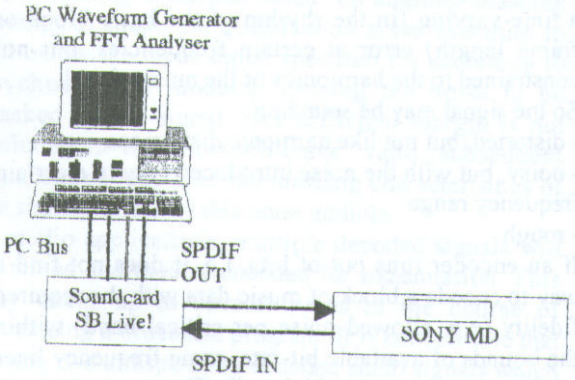


Fig.6 General set-up for digital transfer of objective test waveforms

4.3 Basic Waveform Tests

4.3.1 Test 1

Objective

To illustrate the increased quantisation noise of perceptually masked components and to show how this varies with spectral frequency and amplitude.

Contents of Waveforms

To determine the effect of ATRAC encoding on the masked component of a rudimentary complex waveform, signals consisting of a *masker* component, f_1, and a *maskee*, f_2, were applied. Based on the theory of one critical band being aproximately 1/4 of an octave, waveforms were created consisting of two partials, separated in frequency by one critical band using the following formula:

$$f_{mix}=f_1+f_2 \text{ where } f_2=5/4\ f_1 \qquad (1)$$

Synopsis of results

At lower frequencies, the performance of the ATRAC encoder shows no measurable increase in spectral quantisation noise (see figures 7 to 10). At higher frequencies however, there is a significant increase in spectral quantisation noise, rising by an average of +10dB at 10kHz through to the upper frequency limit as shown in figure 12.

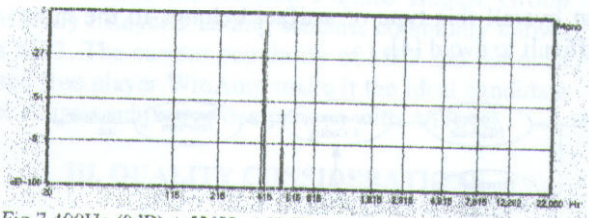


Fig.7 400Hz (0dB) + 520Hz (-50dB) 16bit, 44.1kHz PCM signal

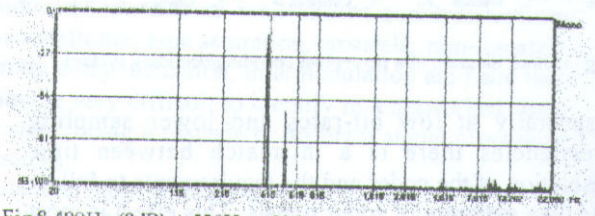


Fig.8 400Hz (0dB) + 520Hz (-50dB) 16bit, 44.1kHz PCM ATRAC encoded signal

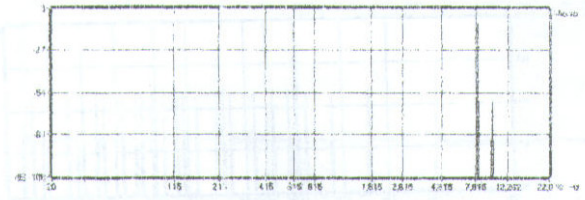


Fig.9 8000Hz (0dB) + 9800Hz (-50dB) 16bit, 44.1kHz PCM signal

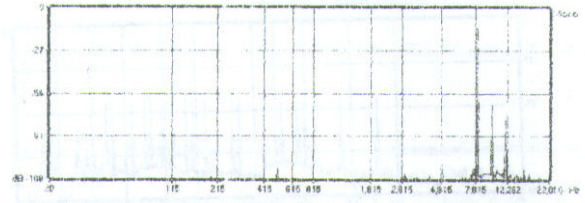


Fig.10 8000Hz (0dB) + 9800Hz (-50dB) 16bit, 44.1kHz PCM ATRAC encoded signal

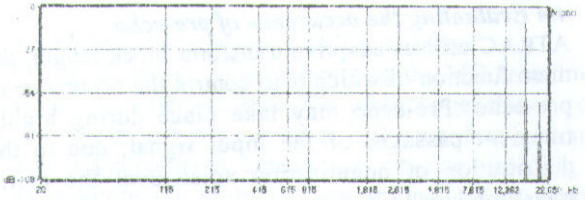


Fig.11 16000Hz (0dB) + 19500Hz (-50dB) 16bit, 44.1kHz PCM signal

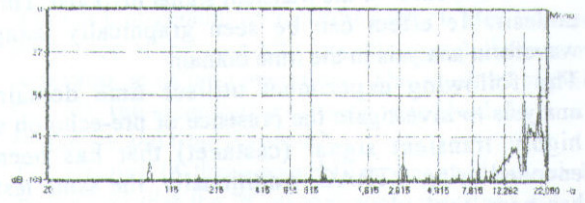


Fig.12 16000Hz (0dB) + 19500Hz (-50dB) 16bit, 44.1kHz PCM ATRAC encoded signal

This possibly reflects the allocation of fixed bits used in the ATRAC bit allocation algorithm, which implements a greater number of fixed bits at lower frequencies i.e. the frequency range where the ear response is most sensitive. The results also suggest that the number of fixed bits at lower frequencies is always greater than the combined number of fixed and variable bits at high frequencies, regardless of the complexity of the signal. Note that the bandwidth of the system under test, when recording simple waveforms, can represent frequencies greater than 19kHz.

4.3.2 Test 2

Objective

To illustrate the reduction in bandwidth and potential for maximum spectral quantisation noise by encoding a signal which is harmonically rich over the entire audible frequency spectrum.

Contents of Waveform

100Hz square wave at 0dB (approx. -4dB full scale digital amplitude)

A square wave consists of a fundamental frequency component, f_1, and an infinite series of odd harmonics e.g. $f_3+f_5+f_7...+f_{odd\infty}$. This abundance of spectral components extends throughout the recoverable bandwidth of a 44.1kHz sample rate recording at considerable levels of amplitude with respect to the fundamental.

As such the application of a square wave can result in a useful indication of the maximum level of increased noise that is likely to occur as a consequence of ATRAC encoding.

Synopsis of results

Two immediate differences occur between the reference (fig.13) signal and the ATRAC encoded signal.

Primarily, a high level of quantisation noise (~+50dB relative to the reference signal) is spread across the entire frequency spectrum as shown in fig.14. Additionally, the bandwidth of the signal is truncated by almost 4kHz at the high end of the spectrum

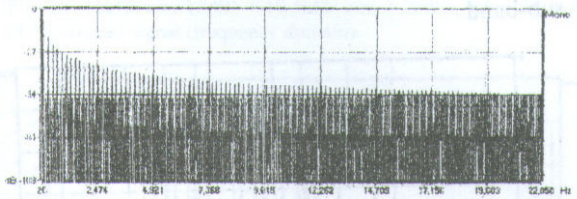


Fig. 13 100Hz PC generated square wave at 0dB, PCM signal

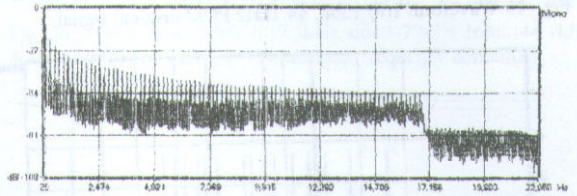


Fig. 14 100Hz PC generated square wave at 0dB, ATRAC encoded signal

This suggest that total number of bits available during the ATRAC bit allocation process is fully required to represent the square wave and the number of bits allocated to the higher extremes of the frequency spectrum is sacrificed to accomodate as much of the preceding bandwidth as possible.

4.3.3 Test 3

Objective

To attempt to demonstrate the allocation of bits performed during ATRAC encoding, with respect to sub-band frequency components and their amplitudes.

Contents of Waveforms

Waveform 1: 400Hz Masking tone (0dB)+150Hz, 250Hz, 355Hz, 450Hz, 560Hz, 720Hz, 1kHz, 1.1kHz, 1.6kHz, 1.9kHz, 2.1kHz, 2.5kHz, 2.9kHz, 3.4kHz, all at -60dB

Waveform 2: 400Hz Masking tone (0dB)+150Hz, 250Hz, 355Hz, 450Hz, 560Hz, 720Hz, 1kHz, 1.1kHz, 1.6kHz, 1.9kHz, 2.1kHz, 2.5kHz, 2.9kHz, 3.4kHz, all at -40dB

Waveform 3: 400Hz Masking tone (0dB)+(150Hz, 250Hz, 355Hz, 450Hz, 560Hz, 720Hz, 1kHz, 1.1kHz, 1.6kHz, 1.9kHz, 2.1kHz, 2.5kHz, 2.9kHz, 3.4kHz, 3.7kHz, 4.4kHz, 5.8kHz, 6.4kHz, 7.7kHz, 9.5kHz, 15.5kHz all at -40dB) + (4.9kHz, 5.8kHz, 8.4kHz, 12.6kHz, 13.5kHz, 17.7kHz all at -60dB)

In addition to differences between adjacent partials of approximately one critical band, the partials are grouped into sub-bands. Waveform 1 contains partials

located in the lower sub-band of frequencies (0-5.5kHz), waveform 2 has the same partials at increased amplitudes and waveform 3 introduces frequency components in the middle and higher sub-bands (5.5kHz +) [5].

Synopsis of results

The spectral components based on the lower sub-band frequencies are recreated with no apparent increase in quantisation noise when at an amplitude of –60dB below the masking tone (fig. 16)

An increase in the amplitude of the masked components of 20dB results in a measurable amount of spectral quantisation noise spread throughout the entire sub-band (fig. 18). This illustrates that the bit allocation algorithm has distributed bits according to the magnitude of the spectral components in the same sub-band.

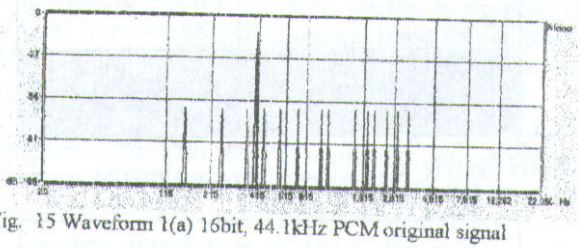


Fig. 15 Waveform 1(a) 16bit, 44.1kHz PCM original signal

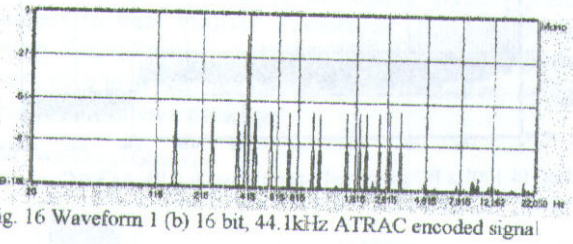


Fig. 16 Waveform 1 (b) 16 bit, 44.1kHz ATRAC encoded signal

The introduction of frequency components in the higher sub-band of frequencies shows a significant increase in spectral quantisation noise, up to 20dB (fig.20). Additionally, the highest frequency component has not been reproduced at all, further illustrating the importance of lower frequency partials during bit allocation process.

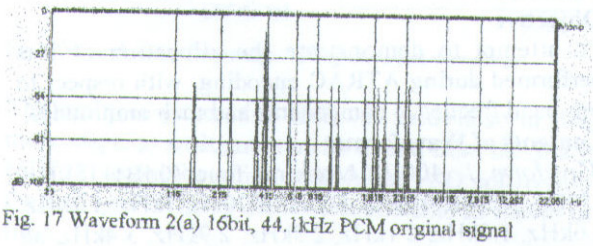


Fig. 17 Waveform 2(a) 16bit, 44.1kHz PCM original signal

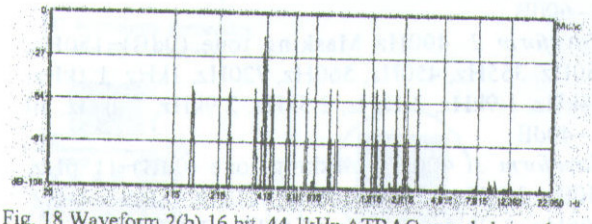


Fig. 18 Waveform 2(b) 16 bit, 44.1kHz ATRAC encoded signal

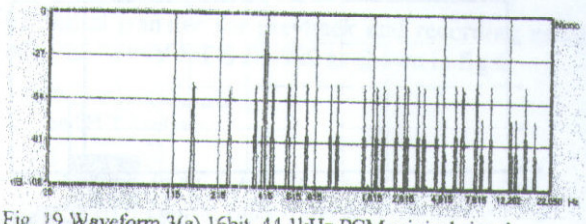


Fig. 19 Waveform 3(a) 16bit, 44.1kHz PCM original signal

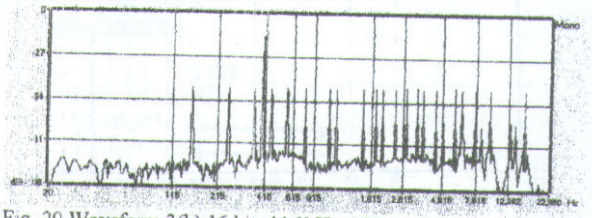


Fig. 20 Waveform 3(b) 16 bit, 44.1kHz ATRAC encoded signal

4.4 Evaluating the occurence of pre-echo

ATRAC utilises adaptive transform block length, the main function of which is to control the occurence of pre-echo. Pre-echo may take place during highly transient passages of the input signal, due to the distribution of quantisation noise over the entire transform block [7].

The audible manifestation of pre-echo is quantisation noise at the onset of the transient signal involved. This undesirable effect can be seen graphically using waveform analysis in the time domain.

The following experiment utilises time domain analysis to investigate the presence of pre-echo on a highly transient signal (castanet) that has been encoded using ATRAC. Additionally, the same test has been applied to a shareware MPEG1 Layer III encoder, at varying bit rates, for comparative measures.

Test 3

Objective

To demonstrate the occurence of pre-echo as a result of encoding a highly transient signal using ATRAC and the Audio Catalyst MP3 encoder at the following bit-rates: 96kbit/s, 128kbit/s and 192kbit/s.

Contents of Waveform

Audio Engineering Society (AES) CD "Perceptual Audio Coders: What to Listen For", track 1 Castanets (fig. 21).

Synopsis of results

ATRAC, with a bit-rate of 292kbit/s, has an immeasurable amount of pre-echo as seen in fig.22. The MP3 encoded signals at 128kbit/s have approximate 150ms of pre-echo signal (fig.23). The superior performance of ATRAC in this test reflects the effectivness of the adaptive transform block size.

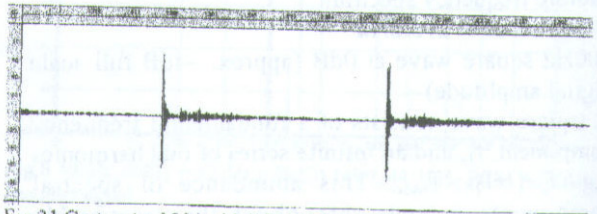


Fig.21 Castanets, 16 bit, 44.1kHz PCM signal

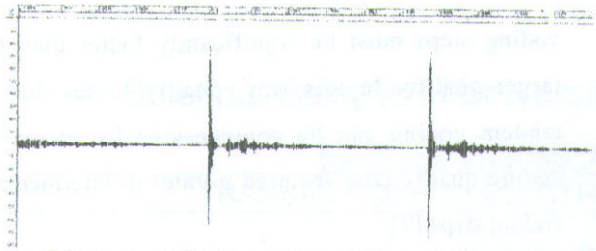


Fig.22 Castanets, ATRAC encoded signal (equiv. 292kbits/s)

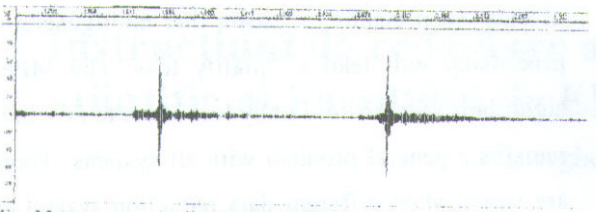


Fig.23 Castanets, Audio Catalyst MP3 Encoder at 128kbits/s

4.5 Multiple Generation ATRAC encoding

One of the major issues concerning the MD recording system is its performance over multiple generations of copying to and from another MD recording system. This is analogous to 'tape to tape' cassette recording.

To execute the multiple generation tests, a single signal was recorded to MD from the computer and the MD recording transferred back to the computer. This was the first generation recording which then become the source of the next MD recording, with the pattern repeating up to ten generations. The resultant signals for all ten generations were analysed in both frequency and time domains.

The creation of these sound files was made possible by the fact that the PC digital audio interface, which was used for repeated transfer of the test waveforms, ignores the copyright flag included in the S/PDIF consumer channel data structure.

Test 4

Objective

To demonstrate, in both the frequency and time domains, the progressive signal degradation due to ATRAC encoding over multiple generations.

Contents of Waveform

16kHz sine wave at 0dB (masking tone) + 19,5kHz sine wave (masked tone) at -70dB relative to the masker [5]

Synopsis of results

The effect of multiple generation recording using ATRAC encoding is shown to result in a progressive degradation of the masking tone level from the first generation (-10dB digital transfer, -16dB analogue transfer). The total level degradation by the tenth generation (digital transfer) is about -24 for the Sharp MD and -18dB for SONY MD. In the case of analogue transfer, after three generation the level dropped down by -30dB.

An interesting observation is that of a level raising of the higher frequency masked spectral component from the original -70dB to about -30dB ÷ -35dB, shown in fig.25. After ten generations this level degrades only with 5dB (-40dB digital transfer).

The increase in spectral quantisation noise by the sixth generation is about 12 dB (fig.27) compared to the PCM original (fig.24)

In addition to the change in spectral content, analysis of later generation waveforms in the time domain displays significant glitching artefacts i.e. signal peaks of excessive amplitude (with respect to the average overall waveform amplitude) that are unrelated, in time placement or frequency, to the original signal content. These are shown in the time domain graphs of figures 26, 28, 30 and 31.

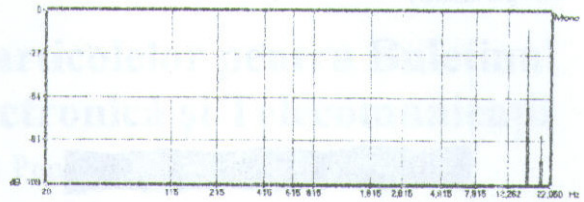


Fig. 24: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz PCM original signal (frequency domain)

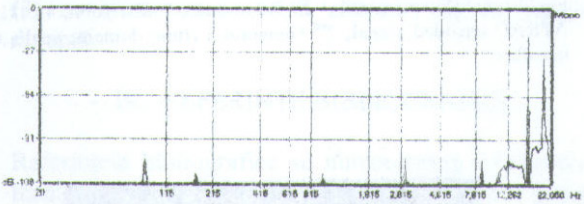


Fig. 25: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 2nd Generation (frequency domain)

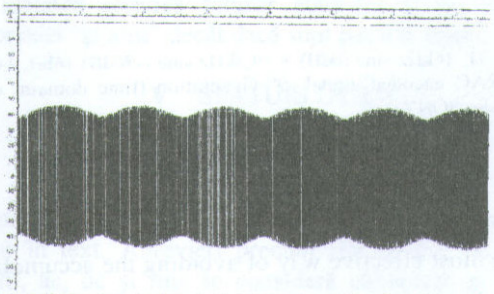


Fig. 26: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 2nd Generation (time domain)

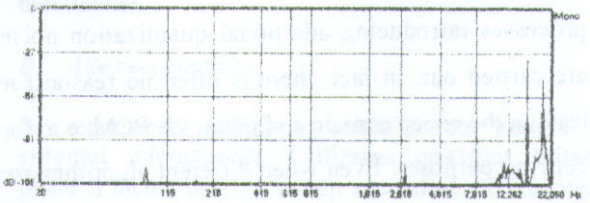


Fig. 27: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 6th Generation (frequency domain)

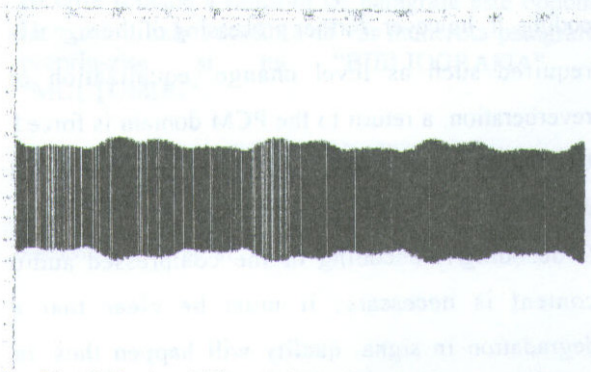


Fig. 28: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 6th Generation (time domain)

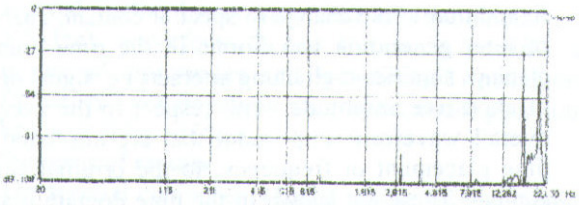


Fig. 29: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 2nd Generation (frequency domain, analogue transfer)

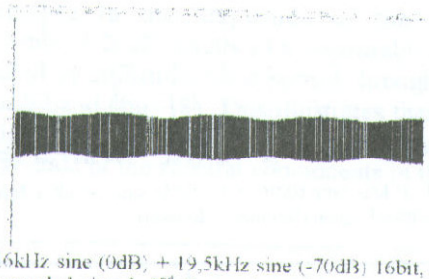


Fig. 30: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 2nd Generation (time domain, analogue transfer)

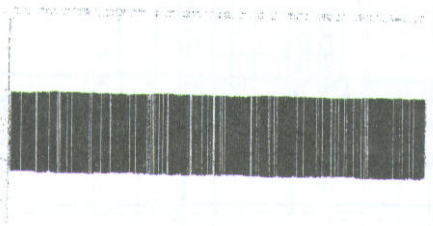


Fig. 31: 16kHz sine (0dB) + 19,5kHz sine (-70dB) 16bit, 44.1kHz ATRAC encoded signal, 9th Generation (time domain, digital transfer SONY)

V. CONCLUSIONS

The most effective way of avoiding the accumulation of errors is to stay within the coded data format as long as possible. Thus no further quantization processes introducing additional quantization noise are carried out. In fact, there is often no reason for leaving the coded domain and going via PCM, e.g. for copying purposes. Even when different algorithm are involved, "transcoding" (the conversion in the coded domain) can probably improve the outcome of tandem coding. If, however, further processing of the signal is required such as level change, equalization or reverberation, a return to the PCM domain is forced. Unfortunately, provisions for interfacing in the coded domain are not yet widely available today.

If decoding/re-encoding of the compressed audio content is necessary, it must be clear that a degradation in signal quality will happen thus, in oreder for the final coded audio to meet a desired target quality, the coding quality at intermediate coding steps must be significantly better than the target quality. In this way, quality losses due to tandem coding can be compensated by increased coding quality (and required bitrate) in intermediate coding steps [7].

Data reduction systems necessarily reduce the amount of data, so it stands to reason that multigeneration processing will lead to quality loss. The MPEG algorithms seem to be slightly less prone to this, but it remains a general problem with all systems. Things are worst when different data reduction system are cascaded, in which case the quality deteriorates very quickly indeed. The symptoms usually become apparent after two or three generations in the form of vague stereo imaging, noise modulation, a harshness to the the sound, aliasing and increased background (quantising) noise.

In general, it is wise to avoid recording a complex mix on a data-reduced if any subsequent processing or copying is envisaged. It would also be sensible to avoid recording stereo source material on data-reduced systems wherever possible, but single mono sources will not be noticeably affected by data reduction.

VI. REFERENCES

[1] R.J. Beaton, "Network Implications of Low Bitrate Audio Codecs", Bericht 18. Tonmeistertagung, Karlsruhe 1994, pp.553

[2] M. Keyhl, J. Herre, C. Schmidmer, "NMR Mesurements on Multiple Generations Audio Coding", presented at the 96th AES Convention 1994 February 26 – March 01, Amsterdam, preprint 3803 (P4.6)

[3] A.J. Mason, A.K. McParland, N.J. Yeadon, - The Atlantic Audio Demostration Equipment, J. Audio Eng. Soc., Vol.48, No.1/2, 2000 January/February, pp.106-118

[4] K. Brandenburg - MP3 and AAC Explained, presented at AES 17th International Conference on High Quality Audio Coding

[5] D. Spector, "Evaluating Adaptive Transform Acoustic Coding for Mini Disc, presented at the 108th AES Convention, 2000 February 19-22, Paris, Preprint nr. 5082 (B-4)

[6] H. Robjohns – "Cutting it Fine", in Sound on Sound, august 1998

[7] M. Erne – "Perceptual Audio Coders – What To Listen For", presented at the 111th AES Convention , 2001 September 21-24, New York, USA

[8] M. Erne – "Digital Audio Compression Algorithms", Proceedings of the First COST-G6 Workshop on Digital Audio Effects, pp.99-110, Barcelona, November 1998

Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 46(60), Fascicola 2, 2001

Instrucţiuni de redactare a articolelor pentru Buletinul ştiinţific al Facultăţii de Electronică şi Telecomunicaţii

Gheorghe I. Popescu[1]

Rezumat – Aceste instrucţiuni sunt concepute să vă prezinte modul de redactare al articolelor pentru Buletinul ştiinţific al Facultăţii de Electronică şi Telecomunicaţii. Prezentul material este conceput ca model pentru modul cum trebuie să arate articolele gata de publicare. Rezumatul trebuie să conţină descrierea problemei, metodele şi soluţiile propuse şi rezultatele experimentale obţinute în cel mult 12 rânduri. Nu se admit referinţe bibliografice în cadrul rezumatului.
Cuvinte cheie: redactare, buletin ştiinţific

I. INTRODUCERE

Formatul buletinului va fi A4. Articolele, inclusiv tabelele şi figurile, nu trebuie să depăşească 6 pagini. Numărul de pagini trebuie să fie obligatoriu par.

II. INSTRUCŢIUNI DE REDACTARE

Articolul trebuie să fie transmis în forma standard descrisă în acest material. Tipărirea se va face cu o imprimantă de bună calitate pe o singură faţă a paginii. Textul se va plasa pe două coloane de 8 cm cu spaţiu de 0,5 cm între ele. Pagina A4 orientată pe înălţimea paginii are marginile de sus şi jos de 1,78 cm, iar cele din stânga şi dreapta de 2,54 cm. Prima pagină a articolului va avea marginea superioară de 5 cm. Pentru editarea articolului se recomandă utilizarea procesorului de text Microsoft Word for Windows cu caractere Times New Roman dactilografiate la un rând. Dimensiunile şi stilul caracterelor sunt: Titlul articolului 18 pt îngroşat, autorul 12 pt rezumatul 9 pt îngroşat, cuvintele cheie 9 pt îngroşat, titlu paragraf 10 pt majuscule, titlu subparagraf 10 pt italic, distanţa de la numărul de ordine la titluri va fi de 0,25 cm, textul normal 10 pt, afilierea autorului 8 pt, notele de subsol 8 pt, legendele figurilor 8 pt şi bibliografia 8 pt.

III. FIGURI ŞI TABELE

Figurile şi tabelele trebuiesc inserate în text aliniate la stânga. Se recomandă evitarea plasării figurilor înainte de prima lor menţionare în text. Se va folosi abrevierea "Fig.1." chiar şi la începutul propoziţiilor. Ecuaţiile trebuie tipărite cu un rând gol deasupra şi dedesubt. Numerotarea lor se face simplu în paranteze: (1), (2), (3) … Numerotarea va fi aliniată faţă de marginea dreaptă.

IV. REFERINŢE BIBLIOGRAFICE

Referinţele bibliografice se numerotează consecutiv în forma [1], [2], [3]… Citările se fac simplu prin plasarea numărului corespunzător [5]. Nu sunt permise referinţe bibliografice în notele de referinţă în subsol. Se recomandă scrierea tuturor autorilor şi nu folosirea expresiei "şi alţii" decât dacă sunt peste 6 autori.

V. SFATURI UTILE

A. Abrevieri şi acronime

Explicitaţi abrevierile şi acronimele prima dată când ele apar în text. Abrevieri precum IEEE, IEE, SI, MKS, CGS, ac, dc şi rms se consideră cunoscute şi nu mai trebuie explicitate. Nu se recomandă utilizarea abrevierilor în titluri decât în cazul când sunt absolut inevitabile.

B. Alte recomandări

Se recomandă utilizarea unităţilor de măsură din sistemul internaţional. Utilizarea unităţilor britanice poate fi făcută doar ca unităţi secundare (în paranteză). Se va evita combinarea unităţilor SI şi CGS. Nu se admit rezultate experimentale preliminare. Numerotarea cu cifre romane a titlurilor de paragrafe este opţională. În cazul utilizării acestora se vor numerota paragrafelor propriu-zise şi nu "BIBLIOGRAFIA" sau "MULŢUMIRI".

BIBLIOGRAFIE

[1] A. Ignea, "Preparation of papers for the International Sympozium Etc. '98", Buletinul Universităţii "Politehnica" Tom 43 (57), 1998, Fascicola 1, 1998, pp. 81

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii Bd. V. Pârvan Timişoara 1900 e-mail geo@ee.utt.ro